TIBCO Adapter™ for Files (IBM i)

User’s Guide

Software Release 4.7.0
June 2010

TIBCO®
The Power of Now®

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 LICENSE.PDF) 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.

TIB, TIBCO, TIBCO Software, TIBCO Adapter, Predictive Business, Information Bus, The Power of Now, TIBCO Enterprise Message Service, TIBCO Rendezvous, TIBCO SmartSockets, TIBCO BusinessWorks, TIBCO Hawk, TIBCO InConcert, and TIBCO Enterprise are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. 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.TXT 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 © 1997-2010 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures . . . vii

Preface . . . ix

Related Documentation . . . x
TIBCO Adapter for Files (IBM i) Documentation. . . . x
Other TIBCO Product Documentation . . . x
How to Contact TIBCO Customer Support . . . xi

Chapter 1 Overview . . . 1

Architecture . . . 2
Transport Mechanisms and Delivery Options . . . 3
Wire Format . . . 3
Functional Components . . . 4
FilePublisher . . . 5
FileSubscriber . . . 8
Message Delivery Considerations . . . 10
Data Transfer Modes . . . 10
Transport Options . . . 10
Comparing ECM and RVCM for Guaranteed Delivery . . . 11
Supported File Types and Data Types . . . 13
BusinessEvents™ Messages . . . 14
JMS Overview . . . 15
Point-to-Point Messaging . . . 15
Publish and Subscribe Messaging . . . 16
Controlling the Flow of Messages. . . . 17
JMS Message Structure . . . 18

Chapter 2 Installation. . . . 21

Installation Requirements . . . 22
System Requirements . . . 22
Storage Requirements . . . 22
Software Requirements . . . 22
Distribution Media and Contents . . . 23
Obtaining the Software . . . 23
Installation Files . . . 23

Uploading the Software 24
Installing the Software 26
Testing the Adapter 28
Sample File Contents 28
Submitting the Initial Sample Using RV 29
Submitting the Initial Sample Using EMS 31
Sample Results 33
Stopping the Adapter 34
Checking the Trace Logs 34
Messages to Operator 35
Uninstalling the Software 36

Chapter 3 Configuring the Adapter 37

Overview 38
Configuration File Structure 38
Configuration Syntax 39
Sample Configuration and Data Files 40
FilePublisher Configuration 43
Trace Section 43
Options Section 47
Pre-Register Section 52
FileType Section 53
Sample Configuration Elements 66
FileSubscriber Configuration 75
Trace Section 75
Options Section 80
FileType Section 90
Sample Configuration Elements 104
Guaranteed Delivery for EMS Messages 111

Chapter 4 Using the Adapter 113

Starting and Stopping the Adapter 114
Starting FilePublisher 114
Starting FileSubscriber 115
Stopping FilePublisher 116
Stopping FileSubscriber 117
Block Transfer Mode 118
File Transfer Using ECM 119
ECM Configuration 120
Sending and Receiving Numeric Data Types 122
Pre-Processing and Post-Processing Files 123

Pre-Processing Files. 123
Post-Processing Files 124
Using Heartbeat Messages 125
Working with Batch Messages. 126
Sending Trigger Messages 127
FilePublisher Usage Guidelines. 128
Sending Numeric Data Types 128
Sending Data Untranslated (OPAQUE) 129
Constructing a Subject Name from Data 129
Publishing Double Values. 130
Pre-registering Subscribers 130
FileSubscriber Usage Guidelines. 131
Receiving Numeric Data Types 131
Adding Header and Trailer Records 131

Appendix A Trace Log Examples 133

Successful FilePublisher Session 134
Failed FilePublisher Session 138
Successful FileSubscriber Session 141
Failed FileSubscriber Session 147

Appendix B Error Messages 153

Error Message Format. 154
Publisher Error Messages 155
SXF0000 – SXF0500 155
SXF1000 – SXF1999 172
SXF2000 – SXF2999 180
SXF3000 – SXF3999 205
SXF4000 – SXF4999 208
Subscriber Error Messages 210
SXF5000 – SXF5999 210
SXF6000 – SXF6999 223
SXF7000 – SXF7999 228
SXF8000 – SXF8999 257
SXF9000 – SXF9999 268

Index 271

Figures

Figure 1 Logical Architecture 2

Figure 2 Functional Components 4

Figure 3 FilePublisher Workflow When Polling. 6

Figure 4 Using a Trigger Subject With a File Name 7

Figure 5 FileSubscriber Workflow. 8

Figure 7 Message Delivery. 15

Figure 8 Point-to-point Message 16

Figure 9 Publish and subscribe messages. 17

Figure 10 Sections of a FilePublisher Configuration File 38

Figure 11 Sections of a FileSubscriber Configuration File 39

Preface

This manual describes how to install, configure, and use the TIBCO Adapter™ for Files (IBM i) software. This manual is primarily intended for users who are involved in setting up and maintaining the system, but may also useful for application programmers.

Topics

- Related Documentation, page x
- How to Contact TIBCO Customer Support, page xi

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Adapter for Files (IBM i) Documentation

The following documents form the TIBCO Adapter for Files (IBM i) documentation set:

- *TIBCO Adapter for Files (IBM i) User's Guide* — Read this document for instructions on using the product.
- *TIBCO Adapter for Files (IBM i) Release Notes* — Read this document for a summary of features, changes since the last release, and a description of any issues that may affect installing or using the Adapter.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:

- TIBCO Rendezvous™ software: This TIBCO's real-time transport layer is supported by the TIBCO Adapter for Files (IBM i) software.
- TIBCO Enterprise Message Service software: This product offers uniform messaging interface among enterprise applications and is supported by the TIBCO Adapter™ for Files (IBM i) software.
- TIBCO Adapter for Files z/OS (MVS): This product allows files on an IBM z/OS system to interoperate with applications configured for the TIBCO environment.

How to Contact TIBCO Customer Support

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- 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:

  http://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.

Chapter 1 Overview

This chapter introduces the TIBCO Adapter for Files (IBM i) by providing background information about features, components, and the application architecture.

Topics

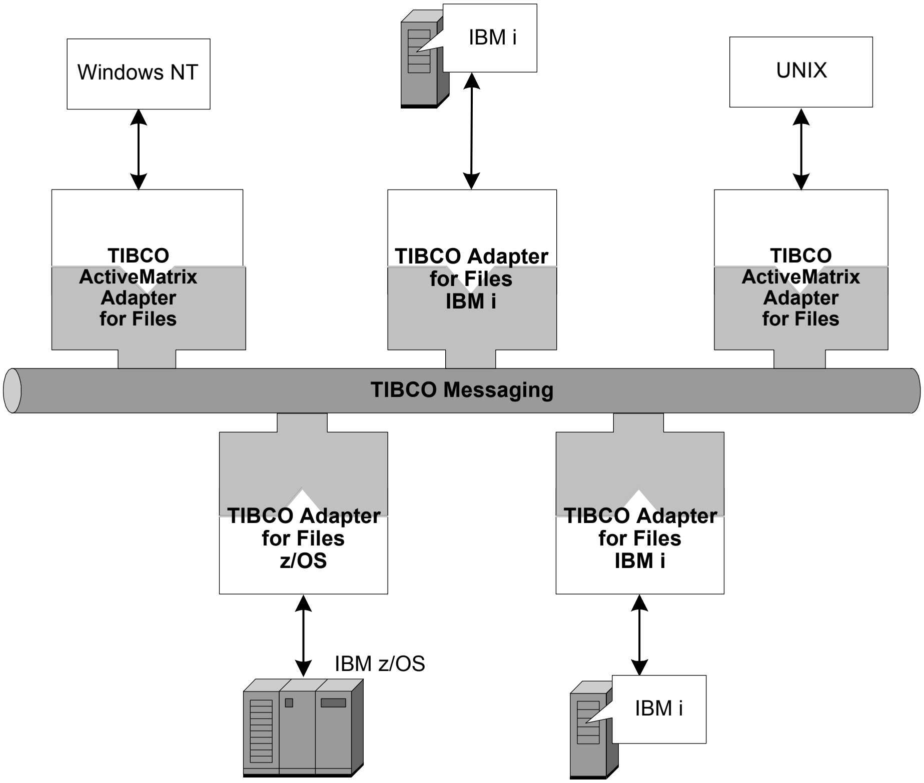
- Architecture, page 2
- Functional Components, page 4
- Message Delivery Considerations, page 10
- Supported File Types and Data Types, page 13
- BusinessEvents™ Messages, page 14

Architecture

TIBCO Adapter for Files (IBM i) is a file handler that supports data files and communicates with other applications enabled for TIBCO Rendezvous or TIBCO EMS.

TIBCO Adapter for Files (IBM i) is typically used in a network operations center or development environment where system administrators, network administrators, and IBM i programmers need to process files between IBM i applications and other operating systems.Typically, the Adapter transfers files from IBM i systems to applications that reside on other operating systems as well as IBM i systems (Figure 1). The Adapter is especially useful in batch-oriented handling of files (for example, passing automated input or output files to a billing system) or for doing quick TIBCO Rendezvousor TIBCO EMS message prototypes.

Figure 1 Logical Architecture


Windows NT
IBM i
UNIX
TIBCO ActiveMatrix Adapter for Files
TIBCO Adapter for Files IBM i
TIBCO ActiveMatrix Adapter for Files
TIBCO Messaging
TIBCO Adapter for Files z/OS
TIBCO Adapter for Files IBM i
IBM z/OS
IBM i

Users with substantial experience in TIBCO messaging formats can use the Adapter to communicate with TIBCO BusinessWorks™. If any transformations are required across platforms, such as EBCDIC to ASCII, or Big Endian to Little Endian, conversions are handled by TIBCO Rendezvous. There are options to add constant fields to a message, and to write constant fields in the output file.

The following import and export subjects are used by ECM for the handshakes between Publisher and Subscriber.

- `_TIBCO_AE_ADAPTER_FAFT_SUBTOPUB.`<*endpoint subject name*>
- `_TIBCO_AE_ADAPTER_FAFT_PUBTOSUB.`<*endpoint subject name*>

Transport Mechanisms and Delivery Options

TIBCO Adapter for Files (IBM i) supports the following transport mechanisms:

- TIBCO EMS
- TIBCO Rendezvous Messaging (RV)
- TIBCO Rendezvous Certified Messaging (RVCM)

Users of RVCM have certified delivery with the overhead of ledger files. For simple file transfer, the Adapter supports ECM (Explicit Confirmation Mode), which supplies FTP-like functionality with error handling and recovery. For additional information, see Message Delivery Considerations on page 10.

Wire Format

TIBCO Adapter for Files (IBM i) supports only TIBCO Rendezvous wire format, not AE wire format. Support is for the `MapMessage` message type only, the sole exception being that nested messages in EMS are nested inside `StreamMessages`.

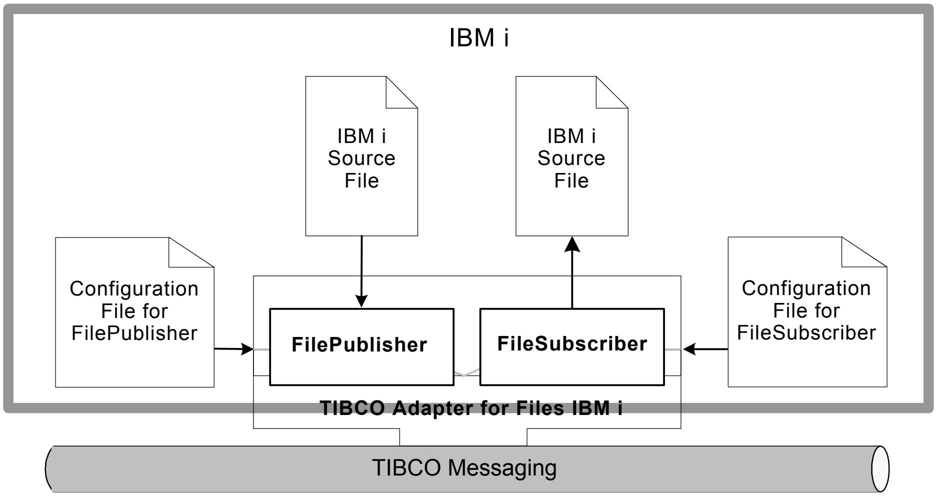
Functional Components

The adapter consists of two applications, as illustrated in Figure 2 below.

- FilePublisher—a publication service that processes data from files and publishes the contents as TIBCO Rendezvous or TIBCO EMS messages.
- FileSubscriber—a subscription service that subscribes to TIBCO Rendezvous or TIBCO EMS messages, processes received messages, and writes the contents to a file.

For each adapter instance, you can configure the format of the incoming or outgoing TIBCO Rendezvous or TIBCO EMS message, the file format, and many file-handling options.

Figure 2 Functional Components


IBM i
IBM i Source File
IBM i Source File
Configuration File for FilePublisher
FilePublisher
FileSubscriber
Configuration File for FileSubscriber
TIBCO Adapter for Files IBM i
TIBCO Messaging

FilePublisher

As an adapter publication service, FilePublisher is defined by specifying the following:

Data Transfer Mode – determines whether FilePublisher transfers data in Block Mode or Record Mode. Each of these modes in turn has a guaranteed mode and a reliable mode. For guaranteed delivery, you can use ECM with the RV message transport, and you can use durable topics for the EMS transport. For details, see Data Transfer Modes on page 10.

File Processing Options – the service can run once, or can be configured to run continuously. When configured for the latter, it processes and publishes files when it receives an event such as a timer event (polling), or an event generated by a message (trigger).

Transport Options – supports the TIBCO Rendezvous, TIBCO Rendezvous Certified Messaging (RVCM), and TIBCO EMS message transport.

Publication Options – in Record Mode of operation, after processing a file, the service provides options to publish the objects generated during the file processing. These options include capabilities for checkpoint restart, to batch objects for publication, delay publication, invoke a callout before publication, or to publish an object only if all of its associations have no errors.

Continuous File Processing

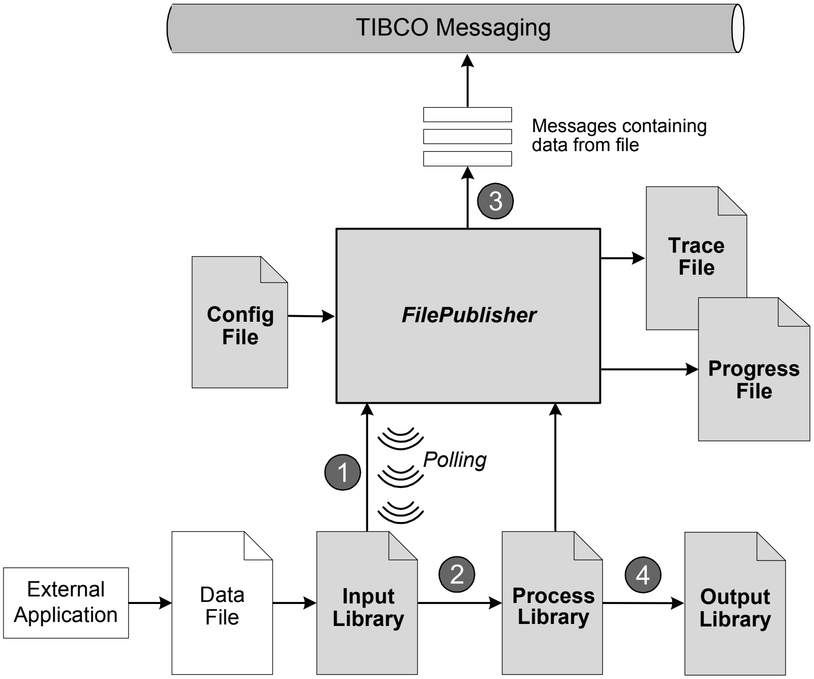
FilePublisher accesses input files either by polling for them or through use of a trigger subject. The polling and trigger-subject methods can operate together. If trigger messages are received while files are being published, the Adapter queues the files and publishes what is in the queue after the current file. A trigger-subject file takes precedence over a polling-method file.

Polling Method

In the polling method, FilePublisher does the following:

1. Selects the file(s) from an input library.
2. Moves the file(s) to a process library.
3. Publishes TIBCO Rendezvous and TIBCO EMS messages from data in the file(s).
4. Moves the file(s) to an output library or removes the file(s) completely.

Figure 3 FilePublisher Workflow When Polling

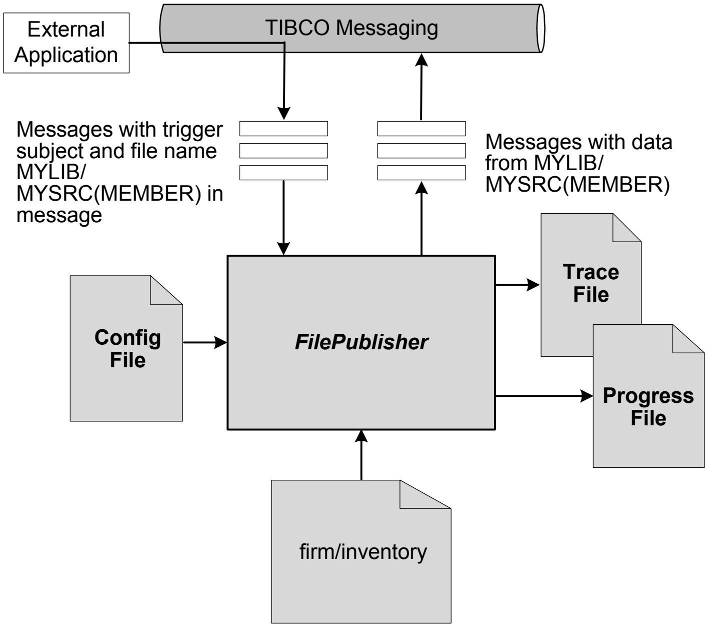


FilePublisher creates a progress file (.PRG) that can be used for viewing the status of file processing. The progress file tracks how far into the file we have published and have received acknowledgements from all subscribers. It is used for recovery.

Trigger-Subject Method

In the trigger-subject method, FilePublisher subscribes to a subject that triggers publishing. If the trigger subject contains a file name in the message, FilePublisher publishes the file as is from its current location. The file is not moved after it is published.

Figure 4 Using a Trigger Subject With a File Name


External Application
TIBCO Messaging
Messages with trigger subject and file name MYLIB/ MYSRC(MEMBER) in message
Messages with data from MYLIB/ MYSRC(MEMBER)
Trace File
Config File
FilePublisher
Progress File
firm/inventory

FileSubscriber

As an adapter subscription service, FileSubscriber is defined by specifying the following:

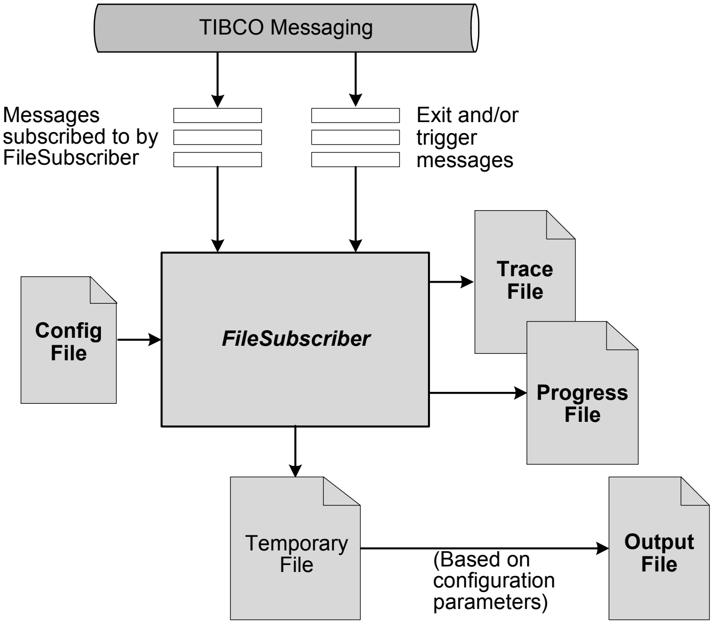
Data Transfer Mode – Determines whether FileSubscriber transfers data in Block Mode or Record Mode. Each of these modes in turn has a guaranteed mode and a reliable mode. For guaranteed delivery, you can use either ECM with the RV message transport or durable topics for the EMS transport. For details, see Data Transfer Modes on page 10.

Processing and File Generation Options – The service can be configured to generate files in a specific library; the service also provides options for executing a command or calling a program before or after generating an output file, and for determining when and how to generate the output file.

Transport Options – supports the TIBCO Rendezvous, TIBCO Rendezvous Certified Messaging (RVCM), and TIBCO EMS message transport (reliable and certified).

Depending on the transport option chosen, FileSubscriber listens to either TIBCO Rendezvous or EMS messages on predefined subjects or destinations and generates output records to a file. FileSubscriber keeps the output in a temporary file and maintains a progress file (Figure 5).

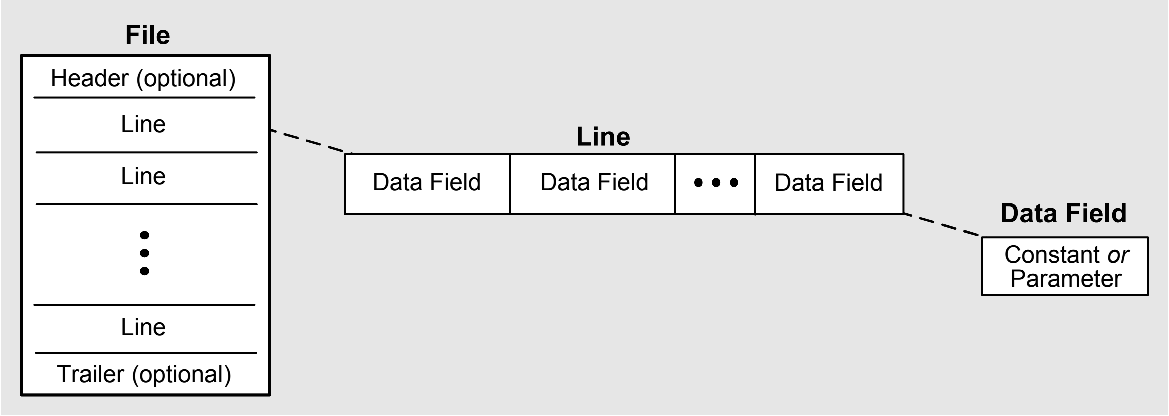
Figure 5 FileSubscriber Workflow


TIBCO Messaging
Messages subscribed to by FileSubscriber
Exit and/or trigger messages
Trace File
Config File
FileSubscriber
Progress File
Temporary File
(Based on configuration parameters)
Output File

Generating Output Files

The format of an output file is shown in Figure 6.

Figure 6 Format of an Output File



When FileSubscriber writes the output file, it uses the `filePrefix` parameter as the output file name. You can also optionally append the system date and time to the file name by using the `appendDateTime` parameter.

Message Delivery Considerations

This section describes the types of message delivery provided by TIBCO Adapter for Files (IBM i).

Data Transfer Modes

TIBCO Adapter for Files (IBM i) allows you to transfer the data in files using the following modes:

- **Block Mode** – the file is read in blocks without regard to its record or field structure. The data are published as blocks and must be subscribed to in block transfer mode as well.
- **Record Mode** – the records and field structure of the data are read and published.

In addition, each of the above data transfer modes has a guaranteed and reliable mode, as follows:

- **Explicit Confirmation Mode (ECM)** – guaranteed mode for RV transport
- **Durable Subscribers** – guaranteed delivery for EMS transport

For additional information, see Block Transfer Mode on page 102.

Transport Options

TIBCO Adapter for Files (IBM i) allows you to publish or subscribe to data using the following message delivery types:

- **Standard Rendezvous reliable delivery (RV)** – reliable message delivery that works well for many programs.
- **Rendezvous certified message delivery (RVCM)** – provides stronger assurances of delivery, along with tighter control, greater flexibility, and fine-grained reporting.
- **TIBCO Enterprise Message Service (EMS)** – provides guaranteed and reliable delivery

For additional information, see *TIBCO Rendezvous Concepts* and *TIBCO Enterprise Message Service User's Guide*.

ECM should not be used with RVCM.

For transfers between the Adapter and TIBCO ActiveMatrix Adapter for Files only block mode over RV is supported with and without ECM.

Comparing ECM and RVCM for Guaranteed Delivery

As noted previously, ECM should not be used with RVCM, and therefore must be used with standard Rendezvous reliable delivery. ECM provides FTP-like functionality with error handling and recovery. While RVCM supports certified messaging, it can be slow when handling large data sets. ECM provides a simplified confirmation mechanism that is suitable for handling large data sets and allows the Adapter to handle data sets of any size.

By default, the Adapter uses RVCM to guarantee message delivery. With RVCM, all messages are held in the publisher's ledger file, along with each subscriber’s state, until they are confirmed by all subscribers. While RVCM guarantees message delivery, its major drawback is that it requires significantly more disk space and memory. RVCM works well in environments that do not require high volume data rates. Its main strength is that it sends messages asynchronously and handles confirmations in the ledger without the application having to deal with it.

In block transfer mode, the Adapter must support high volumes of data (files of 200MB or more), and its performance in transferring files is critical. If RVCM were used to transfer such large data sets, it would require an impractical and unfeasible amount of system resources. Also the data that the Adapter is publishing is already held on disk in the input file, so storing it again in the RVCM ledger adds significantly more overhead.

The following differences between ECM and RVCM are notable:

- With RVCM, slow consumers can cause excessive growth in FilePublisher's memory. In contrast, memory stays constant with ECM. The only trade-off is that all subscribers must confirm each block, since the state is maintained per file type, not per subscriber.
- Since FilePublisher is aware of when the confirmations arrive, it can send the next set of blocks as soon as all the confirmations arrive. This significantly improves the performance of ECM, and it can operate at the speed of the subscriber.

ECM for block transfer requires that each subscriber confirm each block message. The publisher sends out a set of blocks containing file data, and it does not send the next set until it has received confirmations from each subscriber. If the publisher does not receive a block confirmation within a specified time limit, the publisher resends the block message (or messages). This mechanism ensures that the process memory does not to grow based on how fast the subscribers can process the data. Also, because the block data is already persisted in the input file and the publisher (FilePublisher) maintains its state, guaranteed delivery and recovery are achieved.

The following import and export subjects are used by ECM for the handshakes between Publisher and Subscriber:

- `_TIBCO_AE_ADAPTER_FAFT_SUBTOPUB.`<*endpoint subject name*>
- `_TIBCO_AE_ADAPTER_FAFT_PUBTOSUB.`<*endpoint subject name*>

For additional information, see File Transfer Using ECM on page 103.

Supported File Types and Data Types

The Adapter supports a FileType parameter designation of SEQ only. Keyed or random access is not currently supported.

Both the FilePublisher and FileSubscriber support the following numeric (non-text) data types:

- **Zoned Decimal**. The rightmost four bits of a byte are called the numeric bits (N) and normally consist of a code representing a decimal digit. The leftmost four bits of a byte are called the zone bits (Z), except for the rightmost byte of a decimal operand, where these bits may be treated either as a zone or as a sign (S).
- **Binary** (COMP and COMP-4). A binary number occupying 2, 4, or 8 bytes of storage and is handled for arithmetic purposes as a fixed-point number with the leftmost bit being the operational sign.
- **Packed Decimal** (COMP-3). Each byte contains two decimal digits (D), except for the rightmost byte, which contains a sign to the right of a decimal digit.
- **Floating Point – Single (COMP-1)**. A 4-byte single-precision, floating point number. COMP-1 fields do not require a precision definition tag.
- **Floating Point – Double (COMP-2)**. An 8-byte double-precision, floating point number. COMP-2 fields do not require a precision definition tag.

Field Precision and Length

The precision of a field is its length and number of decimals, and is specified using the “precision” tag. The format is `precision="n,p"` where `n` is the number of digits for the number (including decimals) and `p` is the number of decimals.

The length of a field in a record depends on the field’s numeric format and specified precision.

- **Zoned Type** Zoned numbers occupy 1 byte of storage for each numeric digit specified.
- **Binary Type** Binary numbers occupy 2, 4, or 8 bytes of storage, depending on the specified precision: 2 bytes: n from 1 to 4; 4 bytes: n from 5 to 9; and 8 bytes: n from 10 to 18.
- **Packed Type** Packed numbers occupy from 1 to 15 bytes of storage. The number of bytes is determined from the formula n/2 + 1. For example, a field with `precision="7,2"` would occupy 4 bytes of storage.

BusinessEvents™ Messages

TIBCO Adapter for Files (IBM i) can provide business-event level notifications for TIBCO BusinessEvents.

To configure TIBCO Adapter for Files (IBM i) to create business-event messages, include the following element in the Options section of the FileSubscriber and FilePublisher configuration files:

```
EEM_SUBJECT = "destination_subject_name"
```

where *destination_subject_name* is the destination to which the event messages should be routed.

JMS Overview

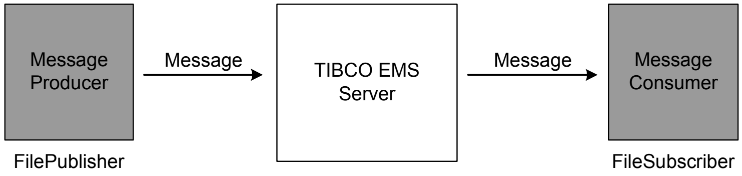
Java Message Service 1.1 (JMS) is a Java framework specification for messaging between applications. Sun Microsystems developed this specification, in conjunction with TIBCO Software and others, to supply a uniform messaging interface among enterprise applications.

Using a message service allows you to integrate the applications within an enterprise. Message-oriented-middleware (MOM) creates a common communication protocol between these applications and allows you to easily integrate new and existing applications in your enterprise computing environment.

JMS is based on creation and delivery of messages. Messages are structured data that one application sends to another. The creator of the message is known as the *producer* and the receiver of the message is known as the *consumer*. The TIBCO EMS server acts as an intermediary for the message and sends it to the correct destination. The server also provides enterprise-class functionality such as fault-tolerance, message routing, and communication with other messaging systems, such as TIBCO Rendezvous™ and TIBCO SmartSockets™.

The diagram below illustrates an application producing a message, sending it by way of the server, and a different application receiving the message.

Figure 7 Message Delivery



JMS supports two messaging models:

- Point-to-point (queues)
- Publish and subscribe (topics)

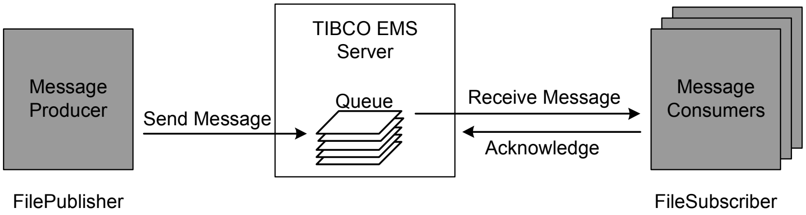
Point-to-Point Messaging

Point-to-point messaging has one producer and one consumer per message. This style of messaging uses a *queue* to store messages until they are received. The message producer sends the message to the queue; the message consumer retrieves messages from the queue and sends acknowledgement that the message was received.

More than one producer can send messages to the same queue, and more than one consumer can retrieve messages from the same queue. The queue can be configured to be exclusive, if desired. If the queue is exclusive, then all queue messages can only be retrieved by the first consumer specified for the queue. Exclusive queues are useful when you want only one application to receive messages for a specific queue. If the queue is not exclusive, any number of receivers can retrieve messages from the queue. Non-exclusive queues are useful for balancing the load of incoming messages across multiple receivers. Regardless of whether the queue is exclusive or not, only one consumer can ever retrieve each message that is placed on the queue.

The diagram below illustrates point-to-point messaging using a non-exclusive queue. Each message consumer receives a message from the queue and acknowledges receipt of the message. The message is taken off the queue so that no other consumer can receive it.

Figure 8 Point-to-point Message



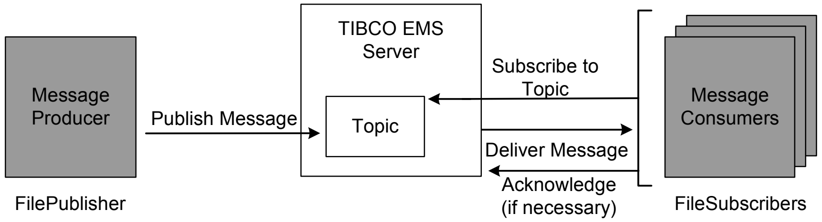
Publish and Subscribe Messaging

In a publish and subscribe message system, producers address messages to a topic. In this model, the producer is known as a *publisher* and the consumer is known as a *subscriber*.

Many publishers can publish to the same topic, and a message from a single publisher can be received by many subscribers. Subscribers subscribe to topics, and all messages published to the topic are received by all subscribers to the topic. This type of message protocol is also known as *broadcast* messaging because messages are sent over the network and received by all interested subscribers, similar to how radio or television signals are broadcast and received.

The diagram below illustrates publish and subscribe messaging. Each message consumer subscribes to a topic. When a message is published to that topic, all subscribed consumers receive the message.

Figure 9 Publish and subscribe messages



There can be a time dependency in the publish and subscribe model. By default, subscribers only receive messages when they are active. If messages are delivered when the subscriber is not available, the subscriber does not receive those messages. JMS specifies a way to remove part of the timing dependency by allowing subscribers to create durable subscriptions. Messages for durable subscriptions are stored on the server until the message expires or the storage limit is reached. Subscribers can receive messages from a durable subscription even if the subscriber was not available when the message was originally delivered.

Controlling the Flow of Messages

You can control the flow of messages to a destination. This is useful when message producers send messages much faster than message consumers can receive them.

JMS Message Structure

JMS messages have a standard structure. This structure includes the following sections:

- Header (required)
- Properties (optional)
- Body (optional)

The JMS specification details a standard format for the header and body of a message. Properties are provider-specific and can include information on specific implementations or enhancements to JMS functionality. The Adapter supports the following property:

Property	Description
`JMS_TIBCO_COMPRESS`	Senders may set this property to request that EMS compress the message before sending it to the server.
`JMS_TIBCO_DISABLE_SENDER`	Senders may set this property to prevent the EMS server from including the sender name in the message when the server sends it to consumers.
`JMS_TIBCO_MSG_TRACE`	Specified in the `FILE_OPTIONS` element of the FileType section of the FilePublisher and FileSubscriber configuration files. Valid values are 'body' and 'null' • When `body` is specified, the entire EMS message is tracked on the EMS server. • When `null` is specified, only the header of the message is tracked. See the EMS documentation for additional information on properties.
`JMS_TIBCO_PRESERVE_UNDELIVERED`	Specifies the message is to be placed on the undelivered message queue if the message must be removed.

The JMS standard specifies two delivery modes for messages, `PERSISTENT` and `NON_PERSISTENT`. TIBCO EMS also includes `RELIABLE_DELIVERY`. This delivery mode eliminates some of the overhead associated with the other delivery modes.

For consumer sessions, you can also specify that consumers do not need to acknowledge receipt of messages, if desired.

More information about properties specific to TIBCO EMS can be found in the *TIBCO Enterprise Message Service Java API Reference*.

Chapter 2 Installation

This chapter explains how to install the TIBCO Adapter for Files (IBM i) software.

Topics

- Installation Requirements, page 22
- Distribution Media and Contents, page 23
- Uploading the Software, page 24
- Installing the Software, page 26
- Testing the Adapter, page 28
- Uninstalling the Software, page 36

Installation Requirements

Before starting the installation procedure, review the topics in this section to verify that your system meets the basic requirements and that you have the prerequisite software installed.

For this release, the product is installed under a new Resource ID and a new product library name. That means that you can leave any previously installed versions of the product as is.

For instructions on uninstalling the current or a previous version of the product, see Uninstalling the Software on page 36.

System Requirements

TIBCO Adapter for Files (IBM i) requires Power Systems server that runs IBM i V5R3 with TCP/IP installed.

Storage Requirements

TIBCO Adapter for Files (IBM i) requires approximately 75 MB of auxiliary storage to install.

Software Requirements

TIBCO Adapter for Files (IBM i) runs in a native IBM i environment. The following software is required and must be installed and configured before starting the installation procedure:

- Operating System — IBM i V5R3 or higher.
- **Optional.** QShell interactive session -- Licensed Program 5722-SS1 Option 30
- The TCP/IP Connectivity.
- A DNS server that can resolve the hostname and host address of the TIBCO Rendezvous and EMS server.

Distribution Media and Contents

This section describes how to obtain the software, and the installation files included in the distribution media. The software is distributed as a *SAVF inside a ZIP file and installed with the RSTLICPGM system command.

Obtaining the Software

To obtain the TIBCO Adapter for Files (IBM i) software, you can download it from the TIBCO Software web site, or you can request a CD.

Downloading from the Web Site

To download the software from the TIBCO web site, follow these steps:

- Contact TIBCO Software Inc. for a password, directory information, and so forth.
- Connect to the TIBCO web site with the required information.
- Download the appropriate ZIP file.

Requesting a CD

If you do not want to download the software over the network, you can obtain a CD that contains the same ZIP file that is obtained from the web. Contact TIBCO Software via email at:

fulfillment@tibco.com

Request a CD for your TIBCO Adapter for Files (IBM i) product installation.

Installation Files

The following ZIP file is included in the distribution media:

`TIB_fa_4.7.0_ibmi_power.zip`

The ZIP file contains these three save files:

`TIBFA470` – contains the *BASE option

`TIBFA470R` - Option 1 for all samples using RV

`TIBFA470E` - Option 2 for all samples using EMS

Uploading the Software

Once you have obtained the ZIP file, you need to upload the software to the IBM i host system using the FTP file transfer utility. Perform the following:

1. Download or copy the `TIB_fa_4.7.0_ibmi._power.zip` file to a PC that can connect to the IBM i.
2. Unzip this file to a temporary location on the PC. It contains the following:
   - `TIBFA470.SAV` – TIBCO Adapter for Files (IBM i) *BASE option
   - `TIBFA470R.SAV` – TIBCO Adapter for Files (IBM i) samples with RV; Option 1
   - `TIBFA470E.SAV` - TIBCO Adapter for Files (IBM i) samples with EMS; Option 2
3. From the temporary location on the PC, open a FTP session with the IBM i:
   ```
   ftp <YOUR_IBMi>
   ```
4. Enter the user ID and password when requested.
5. Specify that the upcoming file transfer locations and names will be specified using IFS notation:
   ```
   quote site namefmt 1
   ```
6. Change the default working directory (library) on the IBM i that will receive the transfer:
   ```
   cd <TARGET_LIBRARY>.LIB
   ```
7. Change the local directory on the PC to the location where the product save files reside:
   ```
   lcd <WINDOWS_DIRECTORY>
   ```
8. Specify a binary transfer:
   ```
   bin
   ```
9. Transfer the `TIBFA470.SAV` file as a `*SAVF` file, and do the same for the `sample` files:
   ```
   PUT TIBFA470.SAV TIBFA470.SAVF
   PUT TIBFA470R.SAV TIBFA470R.SAVF
   PUT TIBFA470E.SAV TIBFA470E.SAVF
   ```

10. Verify that the FTP was successful by signing on to the IBM i as QSECOFR (or a user profile with similar authority) and display the contents of the *SAVF files:

    ```
    DSPSAVF <TARGET_LIBRARY>/TIBFA470
    DSPSAVF <TARGET_LIBRARY>/TIBFA470R
    DSPSAVF <TARGET_LIBRARY>/TIBFA470E
    ```

    This should display a number of objects on the bottom half of the screen.

Installing the Software

Perform the following steps to install the TIBCO Adapter for Files (IBM i) software. All program objects reside within `*SAVF TIBFA470`.

1. Sign-on as `QSECOFR` or as a user profile with similar authority.
2. Use the Restore Licensed Program (`RSTLICPGM`) command to load the software:

   ```
   RSTLICPGM LICPGM(1TIBFA1) DEV(*SAVF)
       SAVF(<TARGET_LIBRARY>/TIBFA470)
   ```

   Parameters:

   — `LICPGM(1TIBFA1)` – specifies the Resource ID of the product being restored.

   — `DEV(*SAVF)` – the device name where the product resides; in this case, the device is a *SAVF whose name is specified in the `SAVF` parameter.

   — `SAVF(<TARGET_LIBRARY>/TIBFA470)` – `*SAVF TIBFA470` resides in library `<TARGET_LIBRARY>`.

3. **Optional.** Use the `RSTLICPGM` command to load the samples that use RV:

   ```
   RSTLICPGM LICPGM(1TIBFA1) DEV(*SAVF) OPTION(1)
       SAVF(<TARGET_LIBRARY>/TIBFA470R)
   ```

   Parameters:

   The same syntax as the preceding step, with `OPTION 1` specified. This parameter specifies which of the optional parts of the licensed program (specified in the `LICPGM` parameter) is to be restored.

4. **Optional.** Use the `RSTLICPGM` command to load the samples that use EMS:

   ```
   RSTLICPGM LICPGM(1TIBFA1) DEV(*SAVF) OPTION(2)
       SAVF(<TARGET_LIBRARY>/TIBFA470E)
   ```

   Parameters:

   The same syntax as the preceding step, with `OPTION 2` specified. This parameter specifies which of the optional parts of the licensed program (specified in the `LICPGM` parameter) is to be restored

After becoming familiar with the product and the samples, you can omit installing the samples on subsequent machines.

The above RSTLICPGM command restores the product components into the appropriate default library names, as follows:

- TIBFALIB for *BASE option
- TIBFASMPRV for Option 1
- TIBFASMPEM for Option 2

Alternatively, you can use other forms of the command to select your own product library name, for example:

```
RSTLICPGM LICPGM(1TIBFA1) DEV(*SAVF) REPLACERLS(*NO)
SAVF(<TARGET_LIBRARY>/TIBFA470)
LIB(<USER-SELECTED-PRODUCT-LIBRARY-NAME>)
```

where `<USER-SELECTED-PRODUCT-LIBRARY-NAME>` is the library name of your choice, for example, `TIBFALIB47`.

Testing the Adapter

Sample configuration files for FilePublisher and FileSubscriber and sample data files for testing the installation are provided in the Adapter installation package. This section explains how to test the Adapter using a sample configuration file.

Sample File Contents

Option 1 of TIBCO Adapter for Files (IBM i) provides the following (in library `TIBFASMPRV`):

- File `INIFPUB` with members `PUB001` through `PUB007`. Each member contains the configuration parameters for the FilePublisher to do a specific type of work. See the `README` file in `TIBFASMPEM` for details, or the prologue at the top of each member.
- File `INIFSUB` with members `SUB001` through `SUB007`. Each member contains the configuration parameters for the FileSubscriber to do a specific type of work. See `README` file in `TIBFASMPEM` for details, or the prologue at the top of each member.
- File `QCLSRC` with members `RUNFA001` through `RUNFA007`. Each member starts up a FilePublisher/FileSubscriber pair. Member `SNDTRG002` is used to send a trigger to `PUB002`.
- The following data files are used in the sample configuration files above: `BIN2`, `FT11`, `FT21`, and `FT31`.
- Backup copies of the data files above: `BKP_BIN2`, `BKP_FT11`, `BKP_FT21`, and `BKP_FT31`.
- Programs `RUNFA001` through `RUNFA007` matching the members in `QCLSRC`.
- Trace log files `PUBLOG` and `SUBLOG`, into which trace logging information can be written.

Each FilePublisher/FileSubscriber pair (except `PUB002/SUB002`) utilizes file polling at timer intervals to query whether there are any *matching* files in `INPUT_LIBRARY` to publish. `PUB002/SUB002` starts publishing when FilePublisher receives an external trigger.

Note that each executing FilePublisher/FileSubscriber pair can be configured to handle multiple FileType definitions concurrently. So all non-conflicting `INIFPUB` file members could be combined into one member, and FilePublisher can process all of these FileType definitions at the same time. The same is true for the Subscriber. For details on FileType definitions, see FileType Section on page 51.

Submitting the Initial Sample Using RV

To submit the initial sample, perform the following:

1. Start TIBCO Rendezvous.

   ```
   SBMJOB CMD(CALL PGM(TIBRV/RVD) PARM('-listen' 'tcp:7500'))
   ALWMLTTHD(*YES)
   ```

   This example assumes that the RV daemon is running on the same machine as the Adapter. If communicating with a RV daemon running on a remote machine, the remote host name has to be specified. For example, `remote_host1:7500`. For additional details, see the TIBCO Rendezvous documentation.

2. Add the following libraries to the library list:

   ```
   ADDLIBLE TIBFASMPRV
   ```

   ```
   ADDLIBLE TIBFALIB
   ```

The publishing process requires a PROCESS_LIBRARY for holding intermediate data and a FilePublisher OUTPUT_LIBRARY for holding the file after it is published. In addition, there must be a FileSubscriber OUTPUT_LIBRARY for holding progress files, intermediate results, and the final published file. These three libraries are required before starting FilePublisher and FileSubscriber.

3. Create the required process and output libraries using the names provided by either of the methods below:

   — Accept the FAPUBPROC, FAPUBOUT, and FASUBOUT library names which are already specified as defaults in the sample configuration files provided.

   or

   — Modify TIBFASMPRV/INIFPUB(PUB001) by changing names provided for PROCESS_LIBRARY and OUTPUT_LIBRARY to libraries of your choice, and modify TIBFASMPRV/INIFSUB(SUB001) by changing the OUTPUT_LIBRARY to a library of your choice.

Before proceeding, note that file FT11 exists in library TIBFASMPRV. If this file is not there because of a previous sample run, duplicate the file from BKP_FT11.

4. Start FilePublisher and FileSubscriber using either of the methods below:

   — Call the sample program RUNFA001.

   ```
   CALL RUNFA001
   ```

or

— Submit the jobs to batch individually from a command entry screen.

```
"SBMJOB     CMD(CALL PGM(SXF3RSUB) PARM('-config'
'TIBFASMPRV/INIFSUB(SUB001)')) ALWMLTTHD(*YES)
```

```
"SBMJOB     CMD(CALL PGM(SXF3RPUB) PARM('-config'
'TIBFASMPRV/INIFPUB(PUB001)')) ALWMLTTHD(*YES)
```

You could also start FilePublisher and FileSubscriber interactively in the QShell environment, as follows:

Display 1:

```
ADDLIBLE TIBFALIB
qsh
cd /usr/tibco/tibfa/PRODUCT-LIBRARY-NAME
sxf3rsub -config 'TIBFASMPRV/inifsub(sub001)'
```

where *PRODUCT-LIBRARY-NAME* is the name of the library in which the product option is installed, usually `TIBFALIB`.

Display 2:

```
ADDLIBLE TIBFALIB
qsh
cd /usr/tibco/tibfa/PRODUCT-LIBRARY-NAME
sxf3rpub -config 'TIBFASMPRV/inifpub(pub001)'
```

where *PRODUCT-LIBRARY-NAME* is the name of the library in which the product option is installed, usually `TIBFALIB`.

When the FilePublisher and FileSubscriber jobs are submitted to batch, ensure that the batch job queues are active and that the programs are running.

Note also that to execute multithreaded applications from QShell, you must set the environment variable `QIBM_MULTI_THREADED`, as follows:

```
ADDENVVAR ENVVAR(QIBM_MULTI_THREADED) VALUE(Y) LEVEL(*SYS)
```

You can also set the environment variable at `LEVEL(*JOB)` so that the variable applies to the current job only.

Submitting the Initial Sample Using EMS

To submit the initial sample, perform the following:

1. Before submitting the samples edit the `EMS_SESSION` parameter in the publisher and subscriber configuration files to reference your EMS Server. For example if your EMS Server is running on a machine with IP address `10.98.32.103` and is using the default port the entry, would look like this:

   ```
   EMS_SESSION    = { providerURL="tcp://10.98.32.19:7222",
   name="EMS_P" }
   ```

   For additional details, see the TIBCO Enterprise Message Service documentation.

2. Add the following libraries to the library list:

   ```
   ADDLIBLE TIBFASMPEM
   ADDLIBLE TIBFALIB
   ```

The publishing process requires a `PROCESS_LIBRARY` for holding intermediate data and a FilePublisher `OUTPUT_LIBRARY` for holding the file after it is published. In addition, there must be a FileSubscriber `OUTPUT_LIBRARY` for holding progress files, intermediate results, and the final published file. These three libraries are required before starting FilePublisher and FileSubscriber.

3. Create the required process and output libraries using the names provided by either of the methods below:

   — Accept the `FAPUBPROC`, `FAPUBOUT`, and `FASUBOUT` library names which are already specified as defaults in the sample configuration files provided.

   or

   — Modify `TIBFASMPEM/INIFPUB(PUB001)` by changing names provided for `PROCESS_LIBRARY` and `OUTPUT_LIBRARY` to libraries of your choice, and modify `TIBFASMPEM/INIFSUB(SUB001)` by changing the `OUTPUT_LIBRARY` to a library of your choice.

Before proceeding, note that file `FT11` exists in library `TIBFASMPEM`. If this file is not there because of a previous sample run, duplicate the file from `BKP_FT11`.

4. Start FilePublisher and FileSubscriber using either of the methods below:

   — Call the sample program `RUNFA001`.

   ```
   CALL RUNFA001
   ```

   or

   — Submit the jobs to batch individually from a command entry screen.

   ```
   "SBMJOB     CMD(CALL PGM(SXF3ESUB) PARM('-config'
   'TIBFASMPEM/INIFSUB(SUB001)')) ALWMLTTHD(*YES)
   ```

   ```
   "SBMJOB     CMD(CALL PGM(SXF3EPUB) PARM('-config'
   'TIBFASMPEM/INIFPUB(PUB001)')) ALWMLTTHD(*YES)
   ```

You could also start FilePublisher and FileSubscriber interactively in the QShell environment, as follows:

Display 1:

```
  qsh
  cd /usr/tibco/tibfa/PRODUCT-LIBRARY-NAME
  sxf3rsub -config 'TIBFASMPEM/inifsub(sub001)'
```

where *PRODUCT-LIBRARY-NAME* is the name of the library in which the product option is installed, usually `TIBFALIB`.

Display 2:

```
   qsh
     cd /usr/tibco/tibfa/PRODUCT-LIBRARY-NAME
     sxf3rpub -config 'TIBFASMPEM/inifpub(pub001)'
```

where *PRODUCT-LIBRARY-NAME* is the name of the library in which the product option is installed, usually `TIBFALIB`.

When the FilePublisher and FileSubscriber jobs are submitted to batch, ensure that the batch job queues are active and that the programs are running.

Note also that to execute multithreaded applications from QShell, you must set the environment variable `QIBM_MULTI_THREADED`, as follows:

```
ADDENVVAR ENVVAR(QIBM_MULTI_THREADED) VALUE(Y) LEVEL(*SYS)
```

You can also set the environment variable at `LEVEL(*JOB)` so that the variable applies to the current job only.

Sample Results

Prior to starting the Adapter, file FT11 was in library TIBFASMPRV or TIBFASMPEM. When the Adapter was started, FilePublisher searched for matches for files with a prefix of FT1 specified in PUB001. File FT11 matched the prefix and was published. The file was copied to the PROCESS_LIBRARY, transmitted to FileSubscriber, and then moved to the FilePublisher OUTPUT_LIBRARY. FileSubscriber received the file and placed it in its OUTPUT_LIBRARY. FilePublisher then continues to poll for files with the prefix FT1.

Stopping the Adapter

FilePublisher and FileSubscriber can be stopped by sending termination messages with appropriate subject names, as shown below (QShell environment):

1. Start QShell:

   ```
   QSH
   ```

2. Change to the following directory:

   ```
   CD /usr/tibco/tibfa/PRODUCT-LIBRARY-NAME
   ```

   where *PRODUCT-LIBRARY-NAME* is the name of the library in which the product is installed, usually `TIBFALIB`.

3. Follow these steps to end the sample:

   a. Send a termination message to end the RV sample:

   ```
   sxftrigr -service 7500 -daemon hawk400.na.tibco.com:7500
   _FILEADAPTER.TEST_FILE_PUB.TERMINATE  'any message   '
   ```

   Make sure the RV daemon parameter is updated to reflect the hostname where the RV daemon is running.

   b. Send a termination message to end the EMS sample:

   ```
   sxftrige -server 'tcp://FA-Dev:7222' '-queue'
   '_FILEADAPTER.GD_FILE_PUB.TERMINATE'  'any message'
   ```

   Make sure the EMS server parameter is updated to reflect the hostname when it resides.

   The termination message contains the name of the FilePublisher or FileSubscriber instance that you wish to stop. For details on Adapter instance names, see the `ADAPTER_NAME` element in either FilePublisher or FileSubscriber configuration in Chapter 3.

Checking the Trace Logs

FilePublisher and FileSubscriber can create trace logs in the files specified by their respective configuration files. You can browse these trace logs for any error, warning, or trace messages.

Examples of trace logs from a successful session and a failed session are shown in Appendix A, Trace Log Examples. Samples PUB001 and SUB001 have been configured to write the trace logs in TIBFASMPRV/PUBLOG(PUB001) and TIBFASMPRV/SUBLOG(SUB001), respectively.

More configuration files and data files are supplied along with adapter software for your reference. For more information about these files see Sample Configuration and Data Files on page 40.

Messages to Operator

Any major check point messages, such as start, stop, and error messages, are sent to QSYSOPR *MSGQ. The command to view the messages is as follows:

```
DSPMSG QSYSOPR
```

Uninstalling the Software

Before uninstalling, you should back up any files that were created during installation and that you have modified, such as configuration files. The uninstallation program deletes all files that were originally created during installation. If you have modified any installed files, they will be removed.

Use the IBM i command `DLTLICPGM` to remove the software from the system:

To delete the current version (4.7), use the `DLTLICPGM LICPGM(1TIBFA1)` command.

To delete the previous version (4.6), use the command `DLTLICPGM LICPGM(1TFA450)`.

Chapter 3 Configuring the Adapter

This chapter describes the choices you have for configuring the Adapter and how to create the required configuration files.

Topics

- Overview, page 38
- FilePublisher Configuration, page 43
- FileSubscriber Configuration, page 72
- Guaranteed Delivery for EMS Messages, page 96

Overview

The section describes the structure of the configuration files, configuration element syntax, and the supported file and data types.

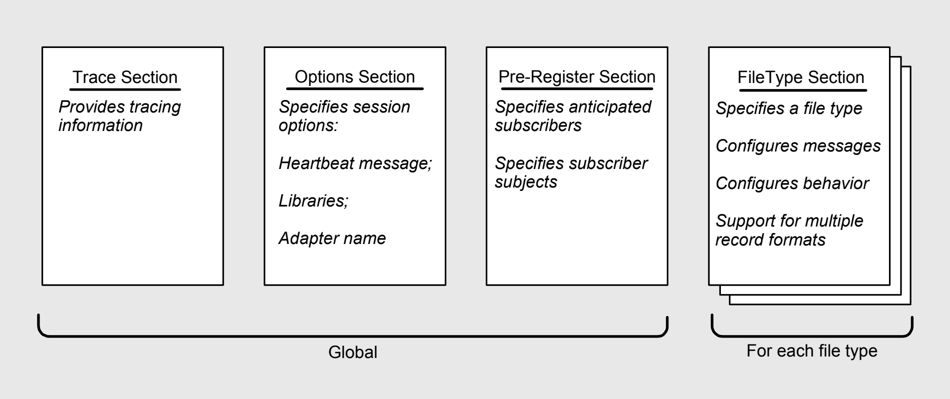
Configuration File Structure

The FilePublisher and FileSubscriber components of TIBCO Adapter for Files (IBM i) have separate configuration files.

FilePublisher

The FilePublisher configuration file consists of the sections shown below (Figure 10):

Figure 10 Sections of a FilePublisher Configuration File

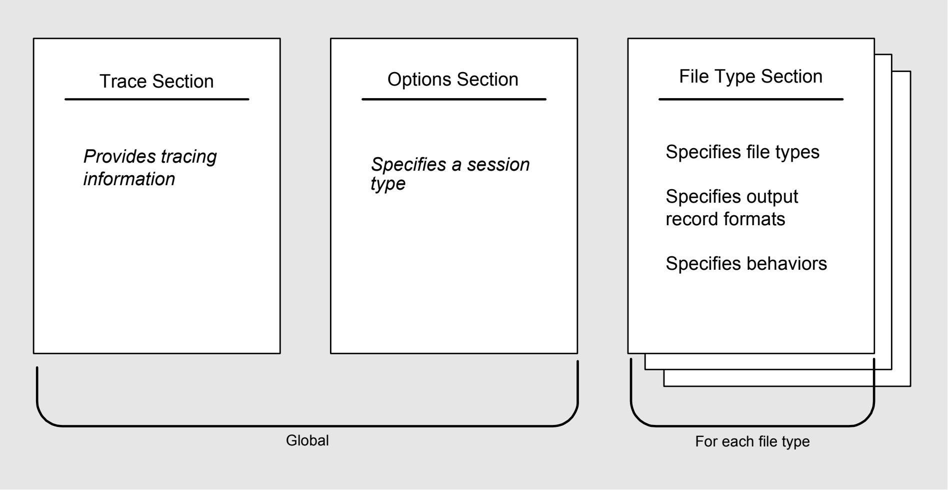


The Trace, Options, and Pre-Register sections contain global configuration values, and there can be one or more FileType sections.

FileSubscriber

The FileSubscriber configuration file consists of the sections shown below (Figure 11).

Figure 11 Sections of a FileSubscriber Configuration File



The Trace and Options sections contain global configuration values, and there can be one or more FileType sections.

Configuration Syntax

Each section of the FilePublisher or FileSubscriber configuration files contain the following organizing entities:

- element – the basic configuration unit. Elements are written in upper case letters.
- parameter – a sub-unit of an element
- tag – a sub-unit of a parameter

Each section consists of elements, and elements consist of parameters surrounded by curly brackets. Each parameter has either a value or consists of tags, surrounded by curly brackets.

For example, the FILE_OPTIONS element from the FileType section shown below consists of parameters with values, separated by commas:

```
FILE_OPTIONS = {   filePrefix="ft1",
                   useFilePolling="true",
                   pollInterval="5000",
                   datasetType=”SEQ”,
                   ...
               }
```

The FILE_LINE element shown below consists of multiple field parameters that have as their value a set of tags with values (separated by commas).

```
FILE_LINE = {
   field = { description = "Part Number", fromMessage="true",
         length = ”6”,type="STRING", value="PartNo", length = ”11”,
         position=”0”},

   field = { description = "Description", fromMessage="true",
         type="STRING", value="Desc", length = ”15”,
         position=”11” },
     ..........}
```

Enclose each element, parameter set, and tag set in curly brackets {}, as illustrated in the examples.

Sample Configuration and Data Files

The Adapter includes both sample configuration files and sample data sets for your use. You have several choices when configuring the Adapter.

- Use the sample configuration files supplied with the Adapter.
- Copy one of the sample configuration files supplied with the Adapter and change it to reflect your particular environment. A detailed discussion of the individual parameters is provided in FilePublisher Configuration on page 43 and FileSubscriber Configuration on page 72.
- Write your own configuration files from scratch. See FilePublisher Configuration on page 43 and FileSubscriber Configuration on page 72 for a discussion of available options and parameters.

Sample Configuration Files

The adapter requires two configuration files to operate: one for FilePublisher and one for FileSubscriber. The sample configuration files and data files in the list that follows are supplied along with adapter software for your reference. They show the usage of many of the features provided by the Adapter software.

Publisher

- PUB001 — Configuration file with example of publishing a delimited text file in Record Mode.
- PUB002 — Configuration file with example of publishing a fixed length text file in Record Mode.
- PUB003 — Configuration file with example of publishing different header and detail formats.
- PUB004 — Configuration file with example of publishing a file with numeric data types, such as packed, zoned, binary, and so on.
- PUB005 — Configuration file with example of publishing a text file in Record Mode with ECM.
- PUB006 — Configuration file with example of publishing a text file in Block Mode irrespective of the record structure.
- PUB007 — Configuration file with example of publishing a text file in Block Mode with ECM.
- PUB008 — Configuration file with example of publishing a text file in Block Mode with Guaranteed EMS delivery feature.
- PUB009 — Configuration file with example of publishing a text file with Record Mode Guaranteed EMS delivery feature.

Subscriber

- SUB001 — Configuration file with example of subscribing and generating a delimited text file in Record Mode.
- SUB002 — Configuration file with example of subscribing and generating a fixed length text file in Record Mode.
- SUB003 — Configuration file with example of subscribing supporting different header and detail formats.
- SUB004 — Configuration file with example of subscribing to a file with numeric data types like packed, zoned, binary, and so on.
- SUB005 — Configuration file with example of subscribing and generating a text file in Record Mode with ECM.
- SUB006 — Configuration file with example of subscribing and generating a text file in Block Mode irrespective of the record structure.
- SUB007 — Configuration file with example of subscribing and generating a text file in Block Mode with ECM.

- SUB008 — Configuration file with example of publishing a text file in Block Mode with Guaranteed EMS delivery feature.
- SUB009 — Configuration file with example of subscribing and generating a textile with Record Mode Guaranteed EMS delivery feature.

Sample Data Files

The following sample data files are included to allow for testing of the Adapter.

- FT11 — Sample data file with delimited text.
- FT21 — Sample data file with fixed length records.
- FT31 — Sample data file with different header and detail formats.
- BIN2 — Sample data file containing numeric data types.

FilePublisher Configuration

This section describes the FilePublisher configuration file. The sections are listed in the order in which the sections appear in the file; the elements that comprise each section are listed in alphabetical order. In addition, sample elements are provided at the end of the section.

Trace Section

You use the Trace section to control the trace logging behavior of FilePublisher. This should be the first section in the configuration file. The elements specified in this section are listed below; all are required (except those used at the request of TIBCO Support).

Element	Description
`FILE_COUNT`	Specifies how many log files to keep. Each saved old log file name has a number (from 1 to the `FILE_COUNT` minus 1) suffixed to it. For example, if `FILE_NAME` is specified as `FPLOG`, and `FILE_COUNT` is set to 5, then there will be four saved log files, named `FPLOG1` through `FPLOG4`, with `FPLOG4` being the earliest (or oldest) file.
`FILE_LIMIT`	Specifies the size of the log file in bytes. When this limit is exceeded, the current log file is saved, and a new log file is created.
`FILE_NAME`	Identifies the log file to be used for trace messages. This is a source physical file. If the file does not exist, then FilePublisher creates one. The default record length is 266 bytes. Default trace log files, `PUBLOG` and `SUBLOG`, are available in TIBFA470R and TIBFA470E depending on which product option(s) were installed. If you require a larger record length, you can create your own file; for example: `CRTSRCPF FILE(GLATONAFA1/TIBLOGNEW) RCDLEN(1000)`
`PRINT_STDOUT`	Specifies whether to send the trace log messages to standard output (`STDOUT`) when set to `true`. This setting is in addition to the tracelog file specified with `FILE_NAME`. If the Adapter jobs are submitted to batch, `STDOUT` will appear in the `QPRINT` printer file. If QSHELL is used to submit Adapter jobs, `STDOUT` will appear on screen.

Element	Description
`TRACE_LEVEL`	Specifies the kind of information that FilePublisher should log. FilePublisher generates trace messages according to the trace level that you specify with this element. • **Trace level 1** generates the following session-level messages: — Initialization of TIBCO Rendezvous, including the version number — TIBCO Rendezvous sessions created — Poll timers and publish trigger subjects set for each file type — Initialization of the TIBCO EMS session created — Publish trigger destinations for each file type • **Trace level 2** generates level 1 messages plus all the values loaded from the config file as below: — All values loaded from the Options section of the configuration file — All values loaded from the Pre-Register section of the configuration file — All values loaded from the FileType sections of the configuration file • **Trace level 3** generates level 1 and 2 messages plus the following field-related messages: — Field processing information (label type, etc.) — Certified Messaging publisher information (name, sequence number, etc.) — Message buffer reallocation — Traces all key I/O Open/Close actions — Traces flow through key publishing modules — Traces flow of re-Publishing requests • **Trace level 4** for detailed tracing. Usually used only at the request of TIBCO Support. — Traces detailed I/O calls for reading/writing data. — Traces detailed processing of Container fields • **Trace level 5** for detailed tracing. Usually used only at the request of TIBCO Support. — Traces all TIBCO RV and EMS received message callbacks

Element	Description
TRACE_LEVEL *(Contd)*	• **Trace Level 6** generates level 1 through 5 messages. Usually used only at the request of TIBCO Support. — Traces all detailed timer callbacks used for handling ECM admin messages, re-publish messages, file lock retry, heartbeat messages, and so on. • **Trace Level 10** generates Level 1 through 6 messages. Usually used only at the request of TIBCO Support. — Detailed trace of all TIBCO RV and EMS message allocates and frees • **Trace Level 15** generates Level 1 through 10 messages. Usually used only at the request of TIBCO Support. — Detailed message content trace (traces the actual contents of the messages, up to 32K bytes)
TRACE_SWITCHES	Special trace switches used by TIBCO support. Do not set this unless explicitly told to by TIBCO support. Valid values are an 8 byte text string. Default is NNNNNNNN.

Options Section

You use the Options section for the following:

- Establish the type of TIBCO Rendezvous or TIBCO EMS session to be used by FilePublisher for publishing
- Name the Adapter instance
- Control heartbeats
- Define certain global elements that apply to all file types defined in the configuration file. These global elements can be overridden in the file type definitions by using the appropriate tags.

Element	Description
ADAPTER_NAME	Specifying this element uniquely identifies the Adapter instance. This element: • Used to set the terminate subject or destination to _FILEADAPTER.<*adapter name*>.TERMINATE. Sending a message to this subject will stop FilePublisher. • Used to set the heartbeat subject or destination to _FILEADAPTER.<*adapter name*>.HEARTBEAT. This is used by the FilePublisher to send heartbeat messages, but does not mean that heartbeats are always published. To publish (send) heartbeats, see PUBLISH_HEARTBEAT.
CONTINUE_ON_ CONFIG_ERROR	Specifies whether to abend the Adapter when it encounters an initialization error for any FileType section. If set to true, and if a configuration error is found for a file, the publisher generates a message to the log and goes on to the next file. If set to false, abend the Adapter. Valid values: true, false. Default is false.
DELETE_PUB_PRG_ FILES	Specifies whether to remove the progress file at the end of a file transfer. Valid values: • none—do not remove the progress file. • single—remove the progress file when it is a 1:1 relationship between the publisher and subscriber. • multiple—remove the progress file. Default is none.

Element	Description
`ECM_NESTED_TEXT_LINES`	Specifies if ECM messages should be sent to the Distributed File Adapter in extended format. Valid values: `true`, `false`. Default is `false`.
`EEM_SUBJECT`	Identifies the subject for messages that the Adapter will pass to TIBCO BusinessEvents. The value is the destination to which the event messages should be routed.
`ERROR_EXIT_CC`	Specifies the return code that FilePublisher returns when exiting with an error condition. Valid values: 4, 8. Default is 8.
`ERROR_SUBJECT`	Specifies where error messages should be published. If this parameter is not blank, all `traceErr` and `FatalErr` messages are copied and sent to that subject. Valid values: blank or a string. Default is blank ““.
`ERROR_DESTINATION`	Specifies where error messages should be published. If this parameter is not blank, all traceErr and FatalErr messages are copied and sent to that destination. Valid values: blank or a string. Default is blank "". **Note**: Specify this field at the end of the Options section. Since the Rendezvous transport or EMS connection is not created until the end of the Options section, any error that occurs can't be published.
`HEARTBEAT_FILE_INFO`	If set to `true`, additional file type information is also published in the heartbeat message. For example, file prefix, file extension, and number of messages published are included. Valid values: `true`, `false`. Default is `true`.
`HEARTBEAT_TIME`	Specifies the interval between heartbeat messages. The value must be greater than or equal to 100 milliseconds. The default value is `60000` milliseconds, or 60 seconds.
`HOST_CODEPAGE`	Works with the codepage support in TIBCO Rendezvous 7.1 and TIBCO EMS 4.2 and above. Specifies the translation table to use on the IBM i side. the CCSID of the job in which the Publisher or Subscriber is running and which is usually tied to the profile of the user who initiated the job.

Element	Description
INPUT_LIBRARY	Default library to search for publishing files.
KILL_PUBLISHER	If set to `true`, enables a publisher to shut down when the Rendezvous daemon shuts down. If not specified, defaults to `false`, which means that the publisher continues to run even if the Rendezvous daemon shuts down. **Note**: This functionality is provided by default in TIBCO EMS. Whenever an exception occurs in a client connection to an EMS server, it is handled by defaul
MAX_CONCURRENT_JOBS	This limits the number of files that will be published at the same time (concurrently). This is used to limit excessive memory and CPU consumption. The default is 12.
NETWORK_CODEPAGE	Works with the codepage support in TIBCO Rendezvous 7.1 or TIBCO EMS 4.2 and above. Specifies the expected codepage sent by any subscribers. Defaults to codepage 819.
OUTPUT_LIBRARY	Default output library to move files to when they have been published using polling.
PRINT_FILE_OPTIONS	If true, prints all configuration information at adapter startup. Valid values: `true`, `false`. Default is true.
PROCESS_LIBRARY	Default process library to be used for holding intermediate and status information while a file is being published.
PUBLISH_HEARTBEAT	A setting to `true` for this element enables publishing of heartbeat messages. If not specified it defaults to `false`, meaning heartbeat messages are not published.
RV_SESSION	Specifying this element along with the following required parameters establishes a reliable mode of publishing: • `name` — A unique alphanumeric name identifying the TIBCO Rendezvous session • `service` — The service group for this session • `network` — Network to initialize a TIBCO Rendezvous session • `daemon` — The name of the TIBCO Rendezvous daemon for this session If you include this element, omit the `RVCM_SESSION` element.

Element	Description
RVCM_SESSION	Specifying this element establishes a certified messaging session. Include all parameters listed for RV_SESSION, plus the following: • ledgerFile — The name of the file-based ledger for Certified Messaging. This file is created in the Integrated File System. • requireOldMessages — Indicates whether a persistent correspondent requires delivery of messages sent to a previous session with the same name for which delivery was not confirmed. A setting to true enables delivery of old unacknowledged messages; a setting to false does not. • defaultTimeLimit — Sets the default message time limit for all outbound certified messages. The time is specified in seconds and the default value is 60. • syncLedger — If you want to use a synchronous ledger file, set this parameter to true. The default for this is false, meaning an asynchronous ledger file is used. If you include this element, omit the RV_SESSION element.
EMS_SESSION	Specifying this element along with the following required parameters establishes reliable mode publishing: • providerURL — The URL of the TIBCO EMS server • name — Unique name for the connection (optional) • username — The user name used to connect to the EMS server • password — The password for the user.
TERMINATE_ON_RV_ SEND_FAIL	Specifies that the Adapter should abend if it cannot successfully call the tibrvsend API. This flag is only applicable to record mode processing. If the file is in block mode, the flag is ignored and the Adapter will be abended. Upon recovery, the Adapter re-syncs to the last check point. In the case of standard block mode, the entire file will be retransmitted. In ECM, retransmission begins with the last block that was successfully acknowledged. Valid values: true, false. Default is false.
WRITE_TO_SYSLOG	If true, major checkpoint log messages are sent to QSYSOPR *MSGQ. Each message includes date and time information. Valid values: true, false. Default is false.

Element	Description
EOL_CHARACTER	Specifies the character that will be used as a delimiter between records in a block. This is necessary when the IBM i platform and the platform that is being targeted do not share a translation table for translating an end-of-line character properly. For example, to communicate the proper line breaks to a Microsoft Windows platform from X'15' to X'0A' there is no system translation table that will do this. This parameter will allow you to specify the character that will translate correctly to X'0A'. Therefore, you would specify `EOL_CHARACTER = "25"`. The `EOL_CHARACTER` is specified as a two-character literal that represents the EBCDIC hexadecimal value.

Pre-Register Section

You use the Pre-Register section to configure and specify the list of anticipated subscribers for the subjects that are published. When the anticipated subscribers are pre-registered, the delivery of Certified Messages is guaranteed regardless of when the subscriber is started, before or after FilePublisher, or when FileSubscriber goes down and is restarted. To use this feature, both FilePublisher and FileSubscriber must establish an `RVCM_SESSION` with the `requireOldMessage` parameter set to `true`.

Not supported for EMS.

Use of this section in your configuration file is optional. If it appears, it should be defined after the Options section. Use the following element to specify the anticipated subscribers for each subject.

Element	Description
SUBSCRIBER	Specify the following required parameters within braces ({ }) for each anticipated subscriber for each subject. Up to 126 `SUBSCRIBER` elements can be defined in the configuration file. • `listenerName` — Specifies the name of the subscribing session. • `listeningSubject` — Specifies the name of the subscription subject.

FileType Section

The FileType section contains two elements:

- FILE_OPTIONS element
- MESSAGE_FIELDS element

FILE_OPTIONS Element

Use the `FILE_OPTIONS` element to specify the following:

- The type of file to publish
- How you want FilePublisher to process the file
- Where the file(s) are located
- How the data in the file is to be accessed
- How the file is to be published

The `FILE_OPTIONS` element is also used to identify those files you want to be published in certified mode if you have started a certified FilePublisher session. A config file can have multiple [FILE OPTIONS] definitions in a single file. See the File PublisherExamples section for details.

The following table lists the parameters in the FILE_OPTIONS elements

Parameter	Description
`filePrefix` (required)	Used to select files from the input library for publishing. While polling, any files with a name that starts with this file prefix are selected for publishing. This prefix can be up to 5 characters. When set to the string `NULL`, and if trigger mode is used, and the full path of the file is specified, then the publisher should not be ignoring this file type. It should generate the temporary file without using the file prefix.
`publishSubjectName` (required)	The subject name to which messages are sent to publish a file.
`publishDestinationName`	Specifies the destination name to which messages are sent to publish a file.
`publishDestinationType`	Specifies the type of the destination to which message are published. For additional details, see JMS Overview on page 15. Valid values: `TOPIC`, `QUEUE`. Default is `TOPIC`.

Parameter	Description
deliveryMode	Specifies the delivery mode for messages. For additional details, see JMS Message Structure on page 18. Valid values: PERSISTENT, NON_PERSISTENT, RELIABLE_DELIVERY. Default is PERSISTENT.
JMS_TIBCO_MSG_TRACE	Specifies whether the entire message, or only the header of the message, is traced. For additional details, see JMS Message Structure on page 18. Valid values: body, null.
EMS_ACK_TYPE	Specifies the acknowledge mode for EMS when durable subscribers are used. Valid values are TIBEMS_AUTO_ACKNOWLEDGE, TIBEMS_CLIENT_ACKNOWLEDGE, TIBEMS_DUPS_OK_ACKNOWLEDGE, TIBEMS_EXPLICIT_CLIENT_ACKNOWLEDGE, TIBEMS_EXPLICIT_CLIENT_DUPS_OK_ACKNOWLEDGE, TIBEMS_NO_ACKNOWLEDGE. Default is TIBEMS_AUTO_ACKNOWLEDGE.

Parameter	Description
Publishing Correlation Identifier	
`useTrackingId`	Specifies whether a GUID-like transaction id is to be used. If the file is configured as needing one, and if no `trackingId` string is received as part of the trigger message, then the publisher automatically generates a GUID-like string for the `trackingId`. If the `trackingId` string comes with the trigger message, the id is forwarded to the subscriber. This information is sent after the publisher receives the final EOF message from the subscriber. Valid values: `true`, `false` (default). If `true`, the following fields are attached to the message that is published on the subject specified by `trackinIdSubject`: • `AdapterName` • `FileName` • `GdgTriggerName          (N/A for IBM i, )` • `FileExtension           (N/A for IBM i, )` • `FilePrefix              (FileType Prefix)` • `FileSize` • `fileTransferDuration` • `DateTime` • `TransferMode            (BLOCK or RECORD)` • `PubLocalResult          (Publisher's Local result = 0 or 8)` • `Result                  (0 or 4 or 8  =  OK or Fail) overall transfer result from Subscriber` • `status                  (Succeeded or Failed)` • `HostName` • `Subject` • `trackingId`
`trackingIdSubject`	Name of the subject to which messages containing the `trackingId` information are published.
Publishing Method (Polling or Trigger Subject)	
`useFilePolling`	If set to `true`, FilePublisher polls for file names to publish from `INPUT_LIBRARY` whose names begin with the `filePrefix` value. A setting to `false` disables file polling. Default is `false`.
`pollInterval`	If file polling is enabled, specifies the time delay between file polls. The time is in milliseconds and the default is 1000.

Parameter	Description
`triggerSubjectName`	Subject to which FilePublisher listens in order to publish a file.
`triggerFieldName`	Use this parameter if the file name is contained in the trigger subject message as a named field.
Message Publishing	
`messagesPerTransaction`	Number of messages to publish at one time (as one transaction). Default is 10.
`transactionDelay`	Acts as a pause between transmission of records or blocks. For ECM, this parameter is used for an additional purpose in conjunction with `retransmissionDelayTicks`. In this case, the publisher will check for any pending confirmations and re-transmit any unacknowledged data messages. See `retransmissionDelayTicks`. Also used to set up the Admin Timeout timer for resending startup messages to the subscribers. The default value is 2000 msec.
`startPublishSubject`	The subject on which FilePublisher sends a message when it starts to publish the file. The published message contains the name of the file that is published.
`endPublishSubject`	The subject on which FilePublisher sends a message after it has completed publishing the file. The end publish message contains the name of the file that is published and the number of messages published. This parameter has a dual purpose. The first is to communicate to the FileSubscriber an end-of-file condition. The second is to communicate that the file has been published. The durable subscribers in EMS require that `endPublishSubject` equal `publishSubject` (or `endPublishDestinationName` equal `publishDestinationName`). Note that publishing an end-of-file indication on a different subject than `publishSubject` in RV may not maintain the correct sequence of messages.

Parameter	Description
generateFileFieldName	Most usages of generateFileSubjectName require that it match the subscribeSubjectName. The Adapter uses the generateFileFieldName parameter as a message differentiator. A message arriving on a subject name can be checked to see whether it is a regular “Data” message, or whether it is a “Generate File” trigger. When an incoming message has a field matching the value specified in the generateFileFieldName parameter, then the message is assumed to be a “Generate File” trigger message, and the subscriber generates the final target output file. The default value for this field is "filename".
Block Transfer Mode	
transferType	Specifies the data transfer mode (i.e, Record Mode or Block Mode) and whether to use ECM or non-ECM with the specified data transfer mode. Valid values: BlockModeECM – use Block Mode with ECM. BlockModeSFT – use Block Mode with non-ECM mode. RecordModeECM – use Record Mode ECM. RecordMode – use Record Mode with non-ECM mode For additional information, see Message Delivery Considerations on page 10.
blockTransferMode	The Adapter reads the file in blocks, without regard to its record or field structure. Valid values: true, false (default).
blockTransferSize	The block size (the number of bytes) the Adapter reads and publishes each time it reads the file. Default is 65536 bytes.
ECM Mode	
useExplicitConfirmation	Use ECM mode. Valid values: true, false (default). If true, block mode is used automatically.

Parameter	Description
`retransmissionDelayTicks`	Only used for ECM Block Mode. This parameter is used to co-ordinate the startup handshake between the Publisher and the Subscriber. For the Publisher, this parameter represents the number of times the `transactionDelay` timer must pop before checking for any unacknowledged Message Blocks. Default value for the Publisher is (`retransmissionDelayTicks*transactionDelay`) sec. Also, this represents the number of times the `transactionDelaytimer` pops before the Publisher attempts to re-synchronize with the Subscriber. For the Subscriber, this parameter represents the number of seconds the FileSubscriber should wait before trying to connect to the Publisher after a startup. Default value for the Subscriber is (`retransmissionDelayTicks * 1000`) sec.
`noWaitAfterConfirmations`	Specifies how a publisher proceeds upon receiving all the acknowledgements from all the ECM subscribers. If set to `true`, the next transaction is processed immediately after FilePublisher receives all the confirmations. If set to `false`, it uses the normal `transactionDelay` parameter to process transactions. Valid values: `true` (default), `false`. Set to `true` for ECM publishers to speed up delivery.
`ECMSubscriberName`	Identifies the ECM subscriber with which this publisher is exchanging acknowledgements in block mode. Repeat this entry for every participating ECM subscriber. This name must be the same as the `ECMSubscriberName` parameter that is specified in the subscriber's configuration file. **Warning**: You cannot specify both an `ECMSubscriberName` and a `confirmationSubject`.
`confirmationSubject`	Confirmation subject name used by FilePublisher to exchange block confirmation messages in record mode. **Warning**: You cannot specify both a `confirmationSubject` and an `ECMSubscriberName`.
File Type and Location	
`dataSetType`	Specifies that only sequentially organized data is allowed. Specify SEQ.

Parameter	Description
`inputLibrary`	Name of the input library. The default is the `INPUT_LIBRARY` specified in the Options section. This overrides the value specified in `INPUT_LIBRARY`.
`processLibrary`	Name of the process libraryfor this file type. The default is the `PROCESS_LIBRARY` specified in the Options section. This overrides the value specified in `PROCESS_LIBRARY`.
`outputLibrary`	Name of the output libraryfor this file type. The default is the `OUTPUT_LIBRARY` specified in the Options section. This is used polling. This overrides the value specified in `OUTPUT_LIBRARY`.
`lineLength`	Record length of the file being published.
Certified Publishing	
`isCertified`	If the FilePublisher session is not a certified session, this parameter is ignored. If the FilePublisher session is specified as a certified session, the default for this parameter is `true`, meaning the file is published in certified mode. If you want to have a specific file published in non-certified (Reliable) mode, you can specify `false`.
`rvcmTimeLimit`	Specify a value for this parameter to override the default value specified in the Options section of the configuration file.
Preprocessing and Postprocessing	
`removeAfterProcess`	If you want to delete a file after it is published, use this parameter. If this parameter is set to true, FilePublisher will delete the file after it is successfully published. The default is false.
`executeBeforeProcess`	Causes FilePublisher to execute a command or call a programbefore the file is published. See Pre-Processing and Post-Processing Files on page 107.
`executeAfterProcess`	Causes FilePublisher to execute a command or call a programafter generating an output file. See Pre-Processing and Post-Processing Files on page 107.
Record Handling	
`removeLeadingBlanks`	Applies to STRING fields. Set this parameter to `true` to have FilePublisher strip leading blanks from a STRING.

Parameter	Description
`removeTrailingBlanks`	Applies to STRING fields. Set this parameter to `true` to have FilePublisher strip trailing blanks.
`keepTrailingBlanks`	If this parameter is set to true, pad records with blanks when publishing a file. Valid values: `true`, `false` (default).
`useFieldWidth`	Identifies the record structure to FilePublisher. If this value is `true`, the fields in the records are assumed to be fixed length.
`delimiter`	Identifies the record as having variable-length fields. The delimiter can be any unique character. If `useFieldWidth` is set to `false`, the default for this field is | (vertical bar).
`startAtLine`	Used by FilePublisher to identify which record from which to start publishing. It can be used to skip over header records in a file. If this parameter is not specified, FilePublisher starts at the first record in the file.
`isBinary`	When set to `true`, specifies that the file data is in binary format. Binary data includes zoned decimal, binary, packed decimal, and floating point data types.
`logMatched`	If the values are set to `true`, these parameters specify that a log file should be created to contain a copy of all the input records which were either matched or unmatched respectively according to the constraint definitions for multi-record-format data. The default value for both the parameters is `false`.
`logUnmatched`	

MESSAGE_FIELDS Element

Use the `MESSAGE_FIELDS` element of the FileType section to configure any messages that are to be published.

- To publish from a file with single record format (either delimited or fixed-width), define one `MESSAGE_FIELDS` element.
- To publish from a file with multi-record format (a file with header and detail records), define one `MESSAGE_FIELDS` element for each record type using the `constraint` parameter.

If you have enabled block transfer mode, the Adapter ignores the `MESSAGE_FIELDS` element.

The MESSAGE_FIELDS element describes the format of the message to be published. It can be formed by using one or more of the following parameters.

Parameter	Description
messageItem	Identifies the field. You can specify different tags inside this parameter. See Tags in the messageItem Parameter on page 59.
messageContainer	Contains message items. See Tags in the messageContainer Parameter on page 61.
constraint	Must be used in the case of multi-record format. See Tags in the constraint Parameter on page 61.

Tags in the messageItem Parameter

The messageItem parameter can identify a field with a constant value, a field from a fixed-width input record, or a field from a delimited input record. In any case, it uses the following tags:

Tag	Description
label	Name of the field.
type	Data type of this field. Valid values: STRING, INTEGER, UNSIGNED INTEGER, SHORT, UNSIGNED SHORT, FLOAT, DOUBLE, BOOLEAN, and TIME. Default is STRING. Valid values for EMS: STRING, INTEGER, SHORT, FLOAT, DOUBLE, BOOLEAN.**Note:** When you specify TIME as the data type, FilePublisher gets the current system time and adds it to the TIBCO message. When FileSubscriber receives the message, it converts the time to a string that represents the time of the publishing system. Specify OPAQUE to send a field untranslated. See Sending Data Untranslated (OPAQUE) on page 113. For COBOL numeric datatypes only, specify one of ZONED, BINARY, PACKED, COMP, COMP-1, COMP-2, COMP-3, or COMP-4 for numeric data. See Sending Numeric Data Types on page 112. **Note:** Make sure isBinary is set to true for both OPAQUE and COBOL numeric data.
precision	For COBOL numeric datatypes only, specifies the size of the field and the number of decimals.

Tag	Description
`convertToString`	For COBOL numeric datatypes only, specifies an alternate conversion. The default value is false, meaning the input numeric field is published as a FLOAT or INTEGER value depending on the type of the field. The datatypes COMP-1, COMP-2, and COMP-3 are published as FLOAT and COMP, COMP-4 as INTEGERs. For a more accurate conversion, specify `true`, meaning a STRING data type will be published.
`value`	Allows you to emulate the COBOL equivalent of `LOW-VALUES` and `HIGH-VALUES`. Valid values: `LOW-VALUES` -- Sets field's value to binary zeros. `HIGH-VALUES` -- Sets field's value to binary `X'FFFFFFFF'`.

For a field with a constant value, add the following tag along with `label` and `type`.

Tag	Description
`constant`	Set this tag to `true` to specify that the field is a constant.
`value`	Use this tag to provide a value for a `constant` field.

For a field from a fixed-width input record, add the following tags along with `label` and `type` to specify the field's position within the record and the length of the field.

Tag	Description
`position`	Identifies the position, starting at 0, of the field within the record. Use `position` *or* `fieldStart` but not both.
`fieldStart`	Used for binary files to specify the start of a field. The first byte of an input or output record is "1." Use `fieldStart` *or* `position` but not both.
`length`	Identifies the length of the field.

For a field from a delimited input record, add the following tag along with `label` and `type` to specify the field's location within the record.

Tag	Description
`location`	Identifies the location of the field within the record, starting with location 1.

Tags in the messageContainer Parameter

You can group message items into containers to form a nested message using the `messageContainer` parameter. Include the following tags along with the `messageItem` or other `messageContainer` that you want to group together.

Tag	Description
`containername`	Name of the container. The name is mandatory. See Example 3, Record Mode Publisher – Different Header and Detail Formats on page 65 for an example of usage.

Tags in the constraint Parameter

The Adapter supports files with multiple record formats. If there is more than one record format, include the `constraint` parameter using the following tags:

Tag	Description
`startNewMessage`	Specifies that this message field definition should start a new message. If this tag is set to `false`, the TIBCO message created by the message field is combined with the previous TIBCO message. The default is `true` for EMS and `false` for RV. Refer to Example 8, Nesting of Mapped Messages, on page 70.
`value`	Specifies the value of the record identifier field. While publishing, this `MESSAGE_FIELD` format is used only if this value matches the record identifier field of the input record.

For a fixed width input record, add the following tags to the `containerName`, `startNewMessage`, and `value` tags.

Tag	Description
`position`	Position, starting at 0, of the record identifier field within the record.
`length`	Length of the field.

For a delimited input record, add the following tag to the `containerName`, `startNewMessage` and `value` tags:

Tag	Description
`location`	Location of the record identifier field within the record, starting with location 1. The example configuration files (Sample Configuration Elements on page 63) illustrate how this tag is used.

Sample Configuration Elements

This section contains several configuration file examples that you can use as models for your own custom configuration file.

Example 1 Record Mode Publisher – Delimited Text File

Configuration file name – `PUB001`, input file – `FT11`.

A portion of the file shows how the data looks:

```
115-01-0500|MONITOR|SONY|VIEWSONIC|01|350.50
115-15-6542|CPU-PIII750MHZ|COMPAQ|PRESARIO|01|900.00
115-67-7356|HDD20GB|SEAGATE|ST500|01|276.60
115-34-8767|FDD54|HP|T24333|01|86.00
115-77-5555|CDRW|HP|T75668|01|350.00
115-78-4646|KEYBOARD|COMPAQ|EASYKEY101|01|32.00
115-88-4454|MOUSE|MICROSOFT|M323|01|26.00
115-36-2727|WINDOWS2000|MICROSOFT|MSW2000|01|400.00
```

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT1",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 useFieldWidth = "false",
                 delimiter = "|",
                 messagesPerTransaction = "10",
                 transactionDelay = "2000",
                 publishSubjectName = "SUB.PUB001.FT1",
                 useFilePolling = "true", pollInterval="5000",
                 removeAfterProcess = "false",
                 transferType = "RecordMode"
               }

MESSAGE_FIELDS = {
messageItem = { location="1", label="PartNo", type="STRING" },
messageItem = { location="2", label="Desc", type="STRING" },
messageItem = { location="3", label="Manufacturer", type="STRING"
},
messageItem = { location="4", label="Model", type="STRING" },
messageItem = { location="5", label="Quantity", type="INTEGER" },
messageItem = { location="6", label="Price", type="FLOAT",
precision="7,2" },
messageItem = { location="7", label="Warranty", type="STRING",
constant="true", value="1 Year" }
}
```

Example 2 Record Mode Publisher – Fixed Length Text File

Configuration file name – `PUB002`, input file – `FT21`.

A portion of the file shows how the data looks:

```
115-01-0500 MONITOR SONY VIEWSONIC 2 250.50
115-15-6542 CPU-PIII750MHZ COMPAQ PRESARIO 04 12.30
115-67-7356 HDD20GB SEAGATE ST500 25 15
115-34-8767 FDD54 HP T24333 1 045.22
115-77-5555 CDRW HP T57213 5 290.00
115-78-4646 KEYBOARD COMPAQ EASYKEY1 10 121.21
```

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT2",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 startAtLine = "1",
                 useFieldWidth = "true",
                 messagesPerTransaction = "5",
                 transactionDelay = "2000",
                 publishSubjectName = "SUB.PUB002.FT2",
                 triggerSubjectName = "TRIGGER.PUB002.FT2",
                 endPublishSubject = "ENDPUB.PUB002.FT2",
                 startPublishSubject = "STARTPUB.PUB002.FT2",
                 useFilePolling = "false", pollInterval="5000",
                 removeAfterProcess = "false",
                 transferType = "RecordMode"
               }

MESSAGE_FIELDS = {
messageItem = { position="0", length = "11", label="PartNo",
type="STRING" },
messageItem = { position="11",length = "15", label="Desc",
type="STRING" },
messageItem = { position="26",length = "11", label="Vendor",
type="STRING" },
messageItem = { position="37",length = "12", label="Model",
type="STRING" },
messageItem = { position="49",length = "2", label="Quantity",
type="INTEGER" },
messageItem = { position="51",length = "8", label="Price",
type="FLOAT" },
messageItem = { position="59", label="Warranty", type="STRING",
constant="true", value="1 Year" }
}
```

Example 3   Record Mode Publisher – Different Header and Detail Formats

As an example of how to use the `constraint` parameter, suppose a file contains both header and detail records that must be combined into one output TIBCO Rendezvous message, but the records have different formats. You could create a TIBCO Rendezvous message as follows:

1. Determine whether the record is a header or detail record. Do this by selecting a field from the record that contains unique data that identifies the field as a header record or as a detail record.
    a. For fixed length files, identify the field by using both the `position` and `length` parameters.
    b. For delimited files, identify the field by using the `location` parameter.
2. Provide the data that identifies the field type. Use the `value` parameter to provide the data.
3. If the record is a header record, specify that TIBCO Rendezvous must start a new message. Do this by specifying the `startNewMessage` parameter to true. For detail records, specify the `startNewMessage` parameter to false if they are to be published with the header record as a single message., otherwise specify `true`, which causes the records to be published as separate messages.
4. The `containerName` parameter is used to identify the container. For this example, the values supplied are `containerName=”Recfm1_Quote”` and `containerName=”Recfm2_Items”`.

Configuration file name – `PUB003`, input file – `FT31`.

A portion of the file shows how the data looks:

```
QUOTE 15004MC001
ITEMS 115-01-0500 MONITOR SONY VIEWSONIC 2 250.50
ITEMS 115-15-6542 CPU-PIII750MHZ COMPAQ PRESARIO 04 12.30
ITEMS 115-67-7356 HDD20GB SEAGATE ST500 25 15
QUOTE 15004MC002
ITEMS 115-34-8767 FDD54 HP T24333 1 045.22
ITEMS 115-77-5555 CDRW HP T57213 5 290.00
ITEMS 115-78-4646 KEYBOARD COMPAQ EASYKEY1 10 121.21
```

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT3",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 startAtLine = "1",
                 useFieldWidth = "true",
                 messagesPerTransaction = "10",
                 transactionDelay = "2000",
                 publishSubjectName = "SUB.PUB003.FT3",
                 useFilePolling = "true", pollInterval="5000",
                 removeAfterProcess = "false",
                transferType = "RecordMode"
              }

MESSAGE_FIELDS = {
constraint = { position="0", length = "5", value="QUOTE",
containerName="Recfm1_Quote",
startNewMessage="true" },
messageItem = { position="5", length = "5", label="QuoteRequestNo",
type="INTEGER" },
messageItem = { position="10", length = "5", label="SupplierCode",
type="STRING" }
}
MESSAGE_FIELDS = {
constraint = { position="0", length = "5", value="ITEMS",
containerName="Recfm2_Items",
startNewMessage="false" },
messageItem = { position="5", length = "11", label="PartNo",
type="STRING" },
messageItem = { position="16",length = "15", label="Desc",
type="STRING" },
messageItem = { position="31",length = "11", label="Vendor",
type="STRING" },
messageItem = { position="42",length = "12", label="Model",
type="STRING" },
messageItem = { position="54",length = "2", label="Quantity",
type="INTEGER" },
messageItem = { position="56",length = "6", label="Price",
type="FLOAT", precision="6,2" },
messageItem = { label="Warranty", type="STRING", constant="true",
value="1 Year" }
}
```

Example 4   Record Mode Publisher - Supporting Numeric Data Types

This example shows a FilePublisher configuration file that supports different numeric data types. It also demonstrates the use of "OPAQUE" keyword. For the FilePublisher sample configuration file, three messageItems are specified.

- The `FILE_OPTIONS` element of the FileType section contains the parameter `isBinary`, specified as `true`.
- The numeric field location is specified using the `position` tag in the `messageItem` parameter. This value specifies the byte number, starting at 1, of the starting position of the field within the record.
- The `precision` tag in the `messageItem` parameter is used to identify the relative length of the field and the number of decimal positions. FilePublisher calculates the length of the field within the record depending on the data type. For example, if the field is identified as `BINARY`, and the precision is specified as `9,2`, then the field occupies 8 bytes of data in the record.

Configuration file name – `PUB004`, input file – `BIN2`.

A portion of the file shows how the data looks:

```
CHAR PACKED       ZONED         BINARY    FLOAT         DOUBLE
abc  1,231,234.12 12,345,678.12 1,234,567 4501.00E-002  123456.7912E+005
def  1,234,567.00 98,989,898.00 1,234,567 15670.10E-001 123456.7000E+001
abc  1,231,234.12 12,345,678.12 1,234,567 14501.00E-002 123456.7912E+005
def  1,234,567.00 98,989,898.00 1,234,567 15670.10E-001 123456.7000E+001
ghi  123.00       999.00        123       12300.00E-002 123123.4567E+002
jkl  123.20       999.11        345,999   34599.90E-001 345999.1121E+004
```

```
[FileType]
FILE_OPTIONS = { filePrefix="BIN2",
                 dataSetType = "SEQ",
                 isBinary = "true",
                 lineLength = "95",
                 startAtLine = "1",
                 useFieldWidth = "true",
                 messagesPerTransaction = "10",
                 transactionDelay = "2000",
                 publishSubjectName = "SUB.PUB004.BIN2",
                 useFilePolling = "true", pollInterval="5000",
                 removeLeadingBlanks = "true",
                 removeTrailingBlanks = "true",
                 removeAfterProcess = "false",
                 transferType = "RecordMode"
               }

MESSAGE_FIELDS = {
messageItem = { fieldStart="01", label="Char", length="5",
type="STRING" },
messageItem = { fieldStart="06", label="Packed", precision="10,2",
convertToString="false", type="PACKED" },
messageItem = { fieldStart="12", label="Zoned", precision="10,2",
convertToString="false", type="ZONED" },
messageItem = { fieldStart="22", label="Binary", precision="10,0",
convertToString="false", type="BINARY" },
messageItem = { fieldStart="30", label="Float", precision="7,2",
length="7", convertToString="false", type="COMP-1" },
messageItem = { fieldStart="34", label="Double", precision="10,4",
length="10", convertToString="false", type="COMP-2" }
}
```

Example 5 Record Mode Publisher with ECM

This example is for publishing a file using ECM. In Record Mode ECM, you need to specify `endPublishSubject` to signal the end of file publishing to the FileSubscriber. The `generateFileSubjectName` parameter in the FileSubscriber should be same as `endPublishSubject`.

Configuration file name – `PUB005`, input file – `FT1`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT1",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 startAtLine = "1",
                 useFieldWidth = "false",
                 delimiter = "|",
                 messagesPerTransaction = "10",
                 transactionDelay = "5000",
                 publishSubjectName = "SUB.PUB005.FT1",
                 endPublishSubject = "ENDPUB.PUB005.FT1",
                 useFilePolling = "true", pollInterval="5000",
                 removeAfterProcess = "false",
                 confirmationSubject = "ECM.PUB005.FT1",
                 totalConfirmationSubscribers = "1",
                 transferType = "RecordModeECM"
}
MESSAGE_FIELDS = {
messageItem = { location="1", label="PartNo", type="STRING" },
messageItem = { location="2", label="Desc", type="STRING" },
messageItem = { location="3", label="Manufacturer", type="STRING"
},
messageItem = { location="4", label="Model", type="STRING" },
messageItem = { location="5", label="Quantity", type="INTEGER" },
messageItem = { location="6", label="Price", type="FLOAT",
precision="7,2" },
messageItem = { location="7", label="Warranty", type="STRING",
constant="true", value="1 Year" }
}
```

Example 6 Block Mode Publisher - Simple File Transfer

You may wish to publish a file without regard to its file structure. To do this, specify that the file is to be published, or subscribed to, in Block Mode SFT.

Configuration file name – `PUB006`, input file – `FT2`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT2",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 startAtLine = "1",
                 messagesPerTransaction = "2",
                 transactionDelay="1000",
                 publishSubjectName = "SUB.PUB006.FT2",
                 endPublishSubject = "ENDPUB.PUB006.FT2",
                 useFilePolling = "true", pollInterval="5000",
                 removeAfterProcess = "false",
                 blockTransferSize = "128000",
                 transferType = "BlockModeSFT"
}
```

Example 7 Block Mode Publisher with ECM

You may wish to publish a file without regard to its file structure and at the same time use ECM. To do this, specify that the file is to be published, or subscribed to, in Block Mode ECM.

Configuration file name – `PUB007`, input file – `FT2`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT2",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 startAtLine = "1",
                 messagesPerTransaction = "2",
                 transactionDelay="2000",
                 publishSubjectName = "SUB.PUB007.FT2",
                 useFilePolling = "true", pollInterval="5000",
                 removeAfterProcess = "false",
                 blockTransferSize = "128000",
                 ECMSubscriberName = "ECM.PUB007.FT2",
                 transferType = "BlockModeECM"
}
```

Example 8 Nesting of Mapped Messages

When nesting map messages, child messages are nested inside a stream message. This allows the nested map messages to be retrieved in the order in which they were inserted from inside the stream message without extra effort. In addition, at each transaction boundary, each map message will identify itself with an internal name-value pair generated by the FilePublisher. This internal name-value pair appears as follows:

```
aXxXCONSTRAINTXxXa={String:dytsth}
```

where `dytsth` is the name defined for the container name in the CONSTRAINT.

Sample MESSAGE_FIELDS definitions:

```
MESSAGE_FIELDS = { CONSTRAINT = {POSITION = "0", length = "1",
                   value = "H", startnewmessage = "true",
                   containername = "dytsth" } ,
   MESSAGEITEM = { position="0", length = "1",
                   label="hdrRec", type="STRING" },
   MESSAGEITEM = { position="1", length = "11",
                   label="hPartNo", type="STRING" },
   MESSAGEITEM = { position="12",length = "15",
                   label="hDesc", type="STRING" },
   MESSAGEITEM = { position="27",length = "11",
                   label="hManufacturer", type="STRING" },
   MESSAGEITEM = { position="38",length = "12",
                   label="hModel", type="STRING" } }

MESSAGE_FIELDS = { CONSTRAINT = {POSITION = "0", length = "1",
                   value = "D", startnewmessage = "false",
                   containername = "dytstd" } ,
   MESSAGEITEM = { position="0", length = "1",
                   label="dtlRec", type="STRING" },
   MESSAGEITEM = { position="1", length = "11",
                   label="PartNo", type="STRING" },
   MESSAGEITEM = { position="12",length = "15",
                   label="Desc", type="STRING" },
   MESSAGEITEM = { position="27",length = "11",
                   label="Manufacturer", type="STRING" },
   MESSAGEITEM = { position="38",length = "12",
                   label="Model", type="STRING" }
   }
```

Sample input data file:

```
Hpartnumber1description1111manufact111modelnum1111
Dpartnumber1description1111manufact111modelnum1111
Hpartnumber2description2222manufact222modelnum2222
Dpartnumber2descriptionXXXXmanufact222modelnum2222
Dpartnumber2description2222manufact222modelnum2222
Hpartnumber3description3333manufact333modelnum3333
Dpartnumber3description3333manufact333modelnum3333
```

The following four messages would be generated and sent by the FilePublisher:

```
StreamMessage={ Header={
JMSMessageID={ID:EMS-SERVER.7B84B0D3513102:25}
JMSDestination={Queue[DY.MRECFM.FILE.BACKUP]} JMSReplyTo={null}
JMSDeliveryMode={PERSISTENT} JMSRedelivered={false}
JMSCorrelationID={null} JMSType={null} JMSTimestamp={Thu Dec 03
10:03:43 EST 2009} JMSExpiration={0} JMSPriority={4} } Properties={
JMS_TIBCO_MSG_EXT={Boolean:true} } Fields={ {MapMsg:{
hDesc={String:description1111} hModel={String:modelnum1111}
aXxXCONSTRAINTXxXa={String:dytsth} hPartNo={String:partnumber1}
hManufacturer={String:manufact111} hdrRec={String:H} }} {MapMsg:{
Manufacturer={String:manufact111} Model={String:modelnum1111}
aXxXCONSTRAINTXxXa={String:dytstd} PartNo={String:partnumber1}
dtlRec={String:D} Desc={String:description1111} }} } }

StreamMessage={ Header={
JMSMessageID={ID:EMS-SERVER.7B84B0D3513102:26}
JMSDestination={Queue[DY.MRECFM.FILE.BACKUP]} JMSReplyTo={null}
JMSDeliveryMode={PERSISTENT} JMSRedelivered={false}
JMSCorrelationID={null} JMSType={null} JMSTimestamp={Thu Dec 03
10:03:44 EST 2009} JMSExpiration={0} JMSPriority={4} } Properties={
JMS_TIBCO_MSG_EXT={Boolean:true} } Fields={ {MapMsg:{
hDesc={String:description2222} hModel={String:modelnum2222}
aXxXCONSTRAINTXxXa={String:dytsth} hPartNo={String:partnumber2}
hManufacturer={String:manufact222} hdrRec={String:H} }} {MapMsg:{
Manufacturer={String:manufact222} Model={String:modelnum2222}
aXxXCONSTRAINTXxXa={String:dytstd} PartNo={String:partnumber2}
dtlRec={String:D} Desc={String:descriptionXXXX} }} {MapMsg:{
Manufacturer={String:manufact222} Model={String:modelnum2222}
aXxXCONSTRAINTXxXa={String:dytstd} PartNo={String:partnumber2}
dtlRec={String:D} Desc={String:description2222} }} } }

StreamMessage={ Header={
JMSMessageID={ID:EMS-SERVER.7B84B0D3513102:27}
JMSDestination={Queue[DY.MRECFM.FILE.BACKUP]} JMSReplyTo={null}
JMSDeliveryMode={PERSISTENT} JMSRedelivered={false}
JMSCorrelationID={null} JMSType={null} JMSTimestamp={Thu Dec 03
10:03:44 EST 2009} JMSExpiration={0} JMSPriority={4} } Properties={
JMS_TIBCO_MSG_EXT={Boolean:true} } Fields={ {MapMsg:{
hDesc={String:description3333} hModel={String:modelnum3333}
aXxXCONSTRAINTXxXa={String:dytsth} hPartNo={String:partnumber3}
hManufacturer={String:manufact333} hdrRec={String:H} }} {MapMsg:{
Manufacturer={String:manufact333} Model={String:modelnum3333}
aXxXCONSTRAINTXxXa={String:dytstd} PartNo={String:partnumber3}
dtlRec={String:D} Desc={String:description3333} }} } }

MapMessage={ Header={
JMSMessageID={ID:EMS-SERVER.7B84B0D3513102:28}
JMSDestination={Queue[DY.MRECFM.FILE.BACKUP]} JMSReplyTo={null}
JMSDeliveryMode={PERSISTENT} JMSRedelivered={false}
JMSCorrelationID={null} JMSType={null} JMSTimestamp={Thu Dec 03
10:03:44 EST 2009} JMSExpiration={0} JMSPriority={4} } Properties={
} Fields={ HOSTNAME={String:IBMI-V5R4.NA.TIBCO.COM}
RecordNumber={Integer:0} filename={String:DY2A} eof={Boolean:true}
numofretries={Integer:1} NumberOfMessages={Integer:3}
trackingid={String:Kq2@@D@@GlPVoE@56azzxkDkzzw}
version={String:5.0} USERID={String:} } }
```

FileSubscriber Configuration

This section describes the FileSubscriber configuration file. The sections are listed in the order in which the sections appear in the file. The elements that comprise each section are listed in alphabetical order. In addition, sample elements are provided at the end of the section.

Trace Section

You use the Trace section to specify the trace logging behavior of FileSubscriber. This should be the first section in the configuration file. The elements specified in this section are listed below; all are required (except those used at the request of TIBCO Support).

Element	Description
FILE_COUNT	Specifies how many log files to keep. Each saved old log file name has a number (from 1 to the FILE_COUNT minus 1) suffixed to it. For example, if FILE_NAME is specified as FPLOG, and FILE_COUNT is set to 5, then there will be four saved log files, named FPLOG1 through FPLOG4, with FPLOG4 being the earliest (or oldest) file.
FILE_LIMIT	Specifies the size of the log file in bytes. When this limit is exceeded, the current log file is saved, and a new log file is created.
FILE_NAME	Identifies the log file to be used for trace messages. This is a source physical file. If the file does not exist, then FilePublisher creates one. The default record length is 266 bytes. Default trace log files, PUBLOG and SUBLOG, are available in TIBFASMPRV or TIBFASMPEM for your use. If you require a larger record length, you can create your own file; for example: CRTSRCPF FILE(GLATONAFA1/TIBLOGNEW) RCDLEN(1000)
PRINT_STDOUT	Specifies whether to send the trace log messages to standard output (STDOUT) of the job in addition to the FILE_NAME specified when set to true. If set to false, trace messages are written only to a log file. If the Adapter jobs are submitted to batch, STDOUT will appear in the QPRINT printer file. If QSHELL is used to submit Adapter jobs, STDOUT will appear on screen.

Element	Description
TRACE_LEVEL	The kind of information that FileSubscriber should log. FileSubscriber generates trace messages according to a trace level (1, 2, or 3) that you specify with this element. • **Trace level 1** generates the following session-level messages: — Initialization of TIBCO Rendezvous, including the version number — TIBCO Rendezvous sessions created — Generate file trigger-subjects set for each file type • **Trace level 2** generates level 1 messages plus all the following values loaded from the configuration file: — All values loaded from the Options section of the configuration file — All values loaded from the FileType sections of the configuration file • **Trace level 3** generates level 1 and 2 messages plus the following field-related messages: — Field processing information (label type, etc.) — Certified Messaging publisher information (name, sequence number, etc.) — Message buffer reallocation — All key I/O Open/Close actions. — Traces flow through key publishing modules. — Traces flow of re-Publishing requests. • **Trace level 4** for detailed tracing. Usually used only at the request of TIBCO Support. — Traces detailed I/O calls for reading/writing data. — Traces detailed processing of Container fields • **Trace level 5** for detailed tracing. Usually used only at the request of TIBCO Support. — Traces all TIBCO RV received message callbacks • **Trace Level 6** generates level 1 through 5 messages. Usually used only at the request of TIBCO Support. — Traces all detailed timer callbacks used for handling ECM admin messages, re-publish messages, file lock retry, heartbeat messages

Element	Description
TRACE_LEVEL *(Contd.)*	- **Trace Level 10** generates Level 1 through 6 messages. Usually used only at the request of TIBCO Support. — Detailed trace of all RV message allocates and frees - **Trace Level 15** generates Level 1 through 10 messages. Usually used only at the request of TIBCO Support. — Detailed message content trace (traces the actual contents of the messages, up to 32K bytes)
TRACE_SWITCHES	Special trace switches used by TIBCO support. Do not set this unless explicitly told to by TIBCO support. Valid values are an 8 byte text string. Default is NNNNNNNN.

Options Section

You use the Options section for the following:

- Establish the type of TIBCO Rendezvous session to be used by FileSubscriber for subscribing
- Name the Adapter instance
- Control heartbeat messages
- Define certain global elements that apply to all file types defined in the configuration file.

You can override these global elements in the FileType section.

Element	Description
ADAPTER_NAME	Uniquely identifies the Adapter instance. This element: - Used to set the terminate subject to _FILEADAPTER.<*adapter name*>.TERMINATE. Sending a message to this subject will either stop the FileSubscriber or the FilePublisher depending on the value of <ADAPTER_NAME>. - Used to set the heartbeat subject to _FILEADAPTER.<*adapter name*>.HEARTBEAT. This is used to send heartbeat messages, but does not mean that heartbeats are published. To publish (or send) heartbeats see PUBLISH_HEARTBEAT.

Element	Description
`CONTINUE_ON_CONFIG_ERROR`	Specifies whether to abend the Adapter when it first encounters an initialization error for any FileType section. If set to `true` and, if a configuration error is found for a file, generate a message to the log and go on to the next file. If set to `false`, abend the Adapter. Valid values: `true`, `false` (default).
`DELETE_SUB_PRG_FILES`	Indicates that the progress file is to be deleted at end of file. Valid values: `true`, `false` (default).
`EEM_SUBJECT`	Identifies the subject for messages that the Adapter will pass to TIBCO BusinessEvents. The value is the destination to which the event messages should be routed.
`EMS_SESSION`	Specifying this element along with the following required parameters establishes reliable mode publishing: • `providerURL` — The URL of the TIBCO EMS server • name — Unique name for the connection (optional) • username — The user name used to connect to the EMS server • password — The password for the user.
`EOL_ON_SUBSCRIPTION`	Specifies the character that will be used as a delimiter between messages and for terminating records.When using hexadecimal, literals can be specified in both lower and uppercase characters. If the required delimiter is the character 'G', the parameter value specified should be 'C7', which is the EBCDIC hex value of 'G'.
`ERROR_EXIT_CC`	Specifies the return code that FileSubscriber returns when exiting with an error condition. Valid values: 4, 8 (default).
`ERROR_SUBJECT`	Specifies where the error message should be published. If this is not blank, all `traceErr` and `FatalErr` messages are copied and sent to that subject. Valid values: blank, string. Default is blank ““.
`HEARTBEAT_FILE_INFO`	Publish additional file type information in the heartbeat message. For example, file prefix, file extension, and number of messages published are included. Valid values: `true` (default), `false`.
`HEARTBEAT_TIME`	Specifies the interval between heartbeat messages. The value must be greater than or equal to 100 milliseconds. The default value is `60000` milliseconds, or 60 seconds.

Element	Description
`HOST_CODEPAGE`	Works with the codepage support in TIBCO Rendezvous 7.1 and above . Specifies the translation table to use on the IBM i mainframe side. Defaults to the CCSID of the job in which the Publisher or Subscriber is running and which is usually tied to the profile of the user who initiated the job.
`NETWORK_CODEPAGE`	Works with the codepage support in TIBCO Rendezvous 7.1 and above. Specifies the expected codepage sent by any remote subscribers.
`OUTPUT_LIBRARY`	Specifies the default output library to be used for all files . This will be suffixed appropriately for the actual output data sets. You can also specify an output data set for each file type. See Configuring the FILE_OPTIONS Element on page 79
`PRINT_FILE_OPTIONS`	Prints all configuration information at adapter startup. Valid values: `true` (default), `false`.
`PUBLISH_HEARTBEAT`	Enables publishing of heartbeat messages. Valid values: `true`, `false` (default). For EMS, heartbeat messages are sent to EMS TOPIC by default.
`QUEUE_LIMIT`	Limits the number of data blocks that the publisher can asynchronously send to prevent over-consumption of memory. This element only supports implementations using SFT (Simple File Transfer) and ECM (Explicit Confirmation Mode). The value is one or more digits specifying the maximum depth of the RV Listen queue for incoming data block messages. The minimum value is 6. The default is 0. **Warning**: `QUEUE_LIMIT should be used` only in ECM or RVCM transport mode. Do not use in RV transport mode. Doing so may result in lost data.

Element	Description
RV_SESSION	If you include this element, **do not** include the RVCM_SESSION element. Specifying this element along with the following required parameters establishes a reliable mode of publishing. • name — Unique alphanumeric name identifying the TIBCO Rendezvous session • service — Service group for this session • network — Network to initialize a TIBCO Rendezvous session • daemon — Name of the TIBCO Rendezvous daemon for this session
RVCM_SESSION	If you include this element, **do not** include the RV_SESSION element. Specifying this element establishes a Certified Messaging session. Include the same parameters as listed for RV_SESSION, plus the following: • ledgerFile — Name of the file-based ledger for Certified Messaging. This file is created in the Integrated File System. • requireOldMessages — Indicates whether a persistent correspondent requires delivery of messages sent to a previous session with the same name for which delivery was not confirmed. Setting this parameter to true enables delivery of old unacknowledged messages; setting it to false does not. • defaultTimeLimit — Sets the default message time limit for all outbound certified messages. The time is specified in seconds and the default value is 60. • syncLedger — If you want to use a synchronous ledger file, set this parameter to true. The default for this is false, meaning an asynchronous ledger file is used.

Element	Description
TERMINATE_ON_RV_SEND_FAIL	Specifies the Adapter should abend if it cannot successfully call the API tibrvsend. This flag is only applicable to record mode processing. If the file is in block mode, the flag will be ignored and the Adapter will be abended. Upon recovery, the Adapter will re-sync to the last check point. In the case of standard block mode, the entire file will be retransmitted. In ECM, retransmission will begin with the last block that was successfully acknowledged. Valid values: `true`, `false`. Default is `false`.
WRITE_TO_SYSLOG	If `true`, major checkpoint log messages are sent to QSYSOPR *MSGQ. Each message includes date and time information. Valid values: `true`, `false`. Default is `false`.

FileType Section

This section describes the file that is to be written. It consists of two elements:

- `FILE_OPTIONS` (See Configuring the FILE_OPTIONS Element on page 79.)
- `FILE_LINE` (See Configuring the FILE_LINE Element on page 86.)

Configuring the FILE_OPTIONS Element

This section describes the parameters in the `FILE_OPTIONS` element. A configuration file can have multiple [FILE_OPTIONS] definitions in a single file.

Parameter	Description
`filePrefix (required)`	Used to construct the name of the file that is written to the output library . Specify a value of up to 5 characters. Also used for deriving the names of the progress file and the work file.
`subscribeSubjectName (required)`	Specifies the subject name to subscribe to for this file type.
Publishing Correlation Identifier	
`useTrackingId`	Specifies that a `trackingId` message is to be sent by the subscriber. This message is created after the subscriber finishes writing the file and receiving the EOF acknowledgement from the publisher. Valid values: `true`, `false` (default). If `true`, the following additional fields are attached to the message published to `trackingIdSubject`: • `AdapterName` • `FileName` • `FileExtension          (N/A for IBM i )` • `FilePrefix             (FileType Prefix)` • `FileSize` • `fileTransferDuration` • `DateTime` • `TransferMode           (BLOCK or RECORD)` • `PubLocalResult        (Publisher's Local result = 0 or 8)` • `Result               (0 or 4 or 8  =  OK or Fail) Overall transfer result from Subscriber` • `status                 (Succeeded or Failed)` • `HostName` • `Subject` • `trackingId`

Parameter	Description
trackingIdSubject	Name of the subject on which messages containing the trackingId information are published.
Data Set Type	
datasetType	Specifies that only sequentially organized data is allowed. Specify SEQ.
Output File Allocation	
outputLibrary	Name of the output library for this file type. The default is what is specified in the OUTPUT_LIBRARY element of the Options section. This overrides the value specified in OUTPUT_LIBRARY.
blockSizeAlloc	Specifies the block size of the subscriber file. Setting this value is highly recommended.
lineLength	Maximum number of characters constituting a line in the output file. This parameter is valid only if skipPadding is true. Setting this value is highly recommended. **Note**: lineLength cannot be larger than 32764 bytes.
truncateRecords	Indicates how the subscriber should behave when the receiving data record length is different from the one specified by the lineLength parameter. Valid values: • wrap—wrap around the rest of the record to a new line. • discard—truncate the record and do not generate an error file • error—truncate the record and rename the working file (WK-prefixed file) to the error file (ERRnn-prefixed file) at EOF (default).
AppendDateTime	If true, FileSubscriber appends the system time to the filePrefix parameter when constructing the name of the file to be written. Default is false.

Parameter	Description
Output File Creation	
autoGenerateFile	If true (default), enables automatic output file creation based on a timer (see the next parameter, saveFileInterval). If false, output files are not generated based on a timer. This value should be explicitly turned off if not required. If autoGenerateFile is set to false, then the saveFileInterval parameter and generateFileOnNumberOfMessages are ignored. When set to false, the file will not be generated until an End-Of-File indication is received from the Publisher.
saveFileInterval	This is the interval of time, in seconds, that the FileSubscriber uses to generate an output file. Valid only if autoGenerateFile is set to true. This parameter is used to periodically save received (staged) data to the target file. Note that the saveFileInterval and generateFileOnNumberOfMessages parameters are mutually exclusive. If the value is 0 (zero) the default value of 120 is used. A maximum value of 1800 is enforced.
generateFileOnNumberOfMessages	Generate an output file if the number of messages received since the generation of the last output file equals this integer value. Default is 0 which means the parameter is not used. This parameter is used to cause the saving of received (staged) data to the target file whenever N number of messages (records) have arrived. The saveFileInterval and generateFileOnNumberOfMessages parameters are mutually exclusive.
noOfRetries	Number of times to try creating the output file. Default is 0.

Parameter	Description
`retryInterval`	The amount of time, in milliseconds, the FileSubscriber waits before retrying when a `File Locked` condition is detected. If value of the `noOfRetries` parameter is greater than `0`, then the value of the `retryInterval` parameter should also be greater than `0`. Default value is 0.
`appendToExistingFile`	If `true`, specifies that if an output file already exists, the data received is appended to the existing file. Otherwise, FileSubscriber overwrites existing data. This option is effective only when `appendDateTime` is set to `false`. **Warning:** If the subscriber is configured with `appendToExistingFile="false"` and if there is an I/O error in writing to the output file, then the Adapter removes the output file first and then renames the work file to error file. Default is `false`.
`exitOnFileSaveError`	Specifies what the Adapter does when it cannot save data to the target file due to insufficient space. If `true`, exit the Adapter. Otherwise continue to accept data and write it to the 'temporary' subscriber file. Valid values: `true`, `false`. Default is `true`.
`forcePublishedFileName`	If this flag is set to '`true`', the output file is named the same as the file being published. Specifically, the output file is named the same as the filename contained in the EOF message. For this parameter to function properly: 1. the `filePrefix` must be specified. 2. `generateFileSubjectName` must be specified **and** must match the `endPublishSubject` as specified in the Publisher's configuration file. Valid values: `true`, `false`. Default is `false`.
`genFilePublishSubject`	If the subject is defined, then the subscriber will publish a message using this subject to announce that a new file is created. It is published after the `executeAfterProcess` is performed.

Parameter	Description
Output Files Based on Trigger Message	
generateFileSubjectName (required)	The subject name to subscribe to for generating the output file for this file type. If autoGenerateFile is set to false, then the saveFileInterval parameter and generateFileOnNumberOfMessages are ignored. When set to false, the file will not be generated until an EOF indication is received from the Publisher.
generateFileFieldName	Most usages of generateFileSubjectName require that it match the subscribeSubjectName. The generateFileFieldName is used to act as a message differentiator, so that the message being sent to a Subscriber can be identified as being a "Generate File" trigger message. It is only required if the subscriber is using the generateFileSubjectName for the associated FileType. Default is "filename".
Certified Subscribing	
If you have specified that the session identified in the configuration file is a certified session, then you can identify those files that you wish to be published in certified or in non-certified (Reliable) mode. **Note:** If you specify a certified session, then all files are published in certified mode unless you specify false for the isCertified parameter.	
isCertified	If the FileSubscriber session is not a certified session, this parameter is ignored. If the FileSubscriber session is specified as a certified session, then the default value for this parameter is true, which means that the file is subscribed to in certified mode. You can specify false if you want to have a specific file subscribed to in non-certified (reliable) mode.
Preprocessing and Postprocessing	
executeBeforeProcess	Causes FileSubscriber to execute a command or call a program before generating an output file. See Pre-Processing and Post-Processing Files on page 107.
executeAfterProcess	Causes FileSubscriber to execute a command or call a program after generating an output file. See Pre-Processing and Post-Processing Files on page 107.

Parameter	Description
Record Handling	
`fileHeader`	Specifies a header record to place at the beginning of the output file.
`fileTrailer`	Specifies a trailer record to place at the end of the output file.
`skipPadding`	If included and set to `true`, specifies that data fields are to be delimited and not padded. In that case, FileSubscriber generates variable length fields in the record. The field position in the output record is determined by the `position` parameter in the `FILE_LINE` element. There is no default value for the parameter delimiter, therefore it is important to specify a value. If `false`, data fields are padded with characters specified by the `padCharacter` parameter and are fixed width. In that case, FileSubscriber generates fixed-length fields in the output record. The file position in the output record is determined by the `position` parameter in the `FILE_LINE` element, and the field length is determined by the `length` parameter.
`padCharacter`	If `skipPadding` is `false`, this alphanumeric character is used as the pad character. Default is a blank character.
`padDirection`	Specifies which direction to pad the data field, `left` or `right`. The default is `right`, which means that the pad characters are added to the right side of the field (that is, left-justified).
`delimiter`	An alphanumeric character used to separate the fields in a line. This parameter is valid only if `skipPadding` is `true`. There is no default, which means that if this parameter is not specified, there is no separator between fields.
`isBinary`	When set to `true`, specifies that the file data is in binary format. Binary data includes zoned decimal, binary, packed decimal, and floating point data types.

Parameter	Description
Transfer Type	
`transferType`	Specifies the data transfer mode (i.e, Record Mode or Block Mode) and whether to use ECM or non-ECM with the specified data transfer mode. Valid values: BlockModeECM – use Block Mode with ECM. BlockModeSFT – use Block Mode with non-ECM mode. RecordModeECM – use Record Mode ECM. RecordMode – use Record Mode with non-ECM mode For additional information, see Message Delivery Considerations on page 10.
`blockTransferMode`	The Adapter writes the data to the file in blocks.Valid values: `true`, `false` (default).
`useExplicitConfirmation`	Use ECM mode. Valid values: `true`, `false` (default).
`ECMSubscriberName`	Name of the corresponding ECM publisher in block mode. This parameter must have a valid value if ECM mode is selected. This parameter may appear more than once in the Publisher with a different value for each Subscriber. **Warning**: You cannot specify both an `ECMSubscriberName` and a `confirmationSubject`.
`SubscriberName`	A new SubscriberName ID is required so that the EMS server can identify which consumer client(s) requires which messages resent after a failure. This SubscriberName ID is created dynamically when the Subscriber is started. For example, SubscriberName="sub1", where sub1 is an identifier of your choice.
`confirmationSubject`	Confirmation subject name used by FileSubscriber to exchange confirmation messages in Record Mode. **Warning**: You cannot specify both a `confirmationSubject` and an `ECMSubscriberName`.

Parameter	Description
`retransmissionDelayTicks`	Only used for ECM Block Mode. This parameter is used to co-ordinate the startup handshake between the Publisher and the Subscriber. For the Publisher, this parameter represents the number of times the `transactionDelay` timer must pop before checking for any unacknowledged Message Blocks (retransmissionDelayTicks * transactionDelay). Default value for the Publisher is 2 sec. Also, this represents the number of times the transactionDelayTimer pops before the Publisher attempts to re-synchronize with the Subscriber. For the Subscriber, this parameter represents the number of seconds the FileSubscriber should wait before trying to connect to the Publisher after a startup (retransmissionDelayTicks * 1000). Default value for the Subscriber is 10 sec.

Configuring the FILE_LINE Element

Use the `FILE_LINE` element in the FileType section to format the output record. If output records are in a different format, you can include multiple `FILE_LINE` elements in a single FileType section. The `FILE_LINE` element contains parameters that are defined using the following tags. These tags are optional unless specified otherwise.

The `FILE_LINE` element describes the format of the message to be published. It can be formed by using one or more of the following parameters.

Parameter	Description
`field`	Identifies the field. You can specify different tags inside this parameter. See Tags in the field parameter on page 87.
`constraint`	Must be used in the case of multi-record format. See Tags in the constraint Parameter on page 88

Tags in the field parameter

The FILE_LINE element's `field` parameter has the following tags.

Tag	Description
`description`	A concise description of the data field. FileSubscriber does not process this parameter; it is treated as a comment.
`fromMessage`	A setting of `true` flags this field as being generated from a data item in the incoming TIBCO Rendezvous message. A setting of `false` means that this field is a constant field. Default is `false`.
`position`	For delimited files, specifies the field position in the record, starting at 1. Otherwise, specifies the byte index in the record, starting at 0. Also see the `skipPadding` parameter for the effect of `position` in output records. Use `position` *or* `fieldStart` but not both.
`fieldStart`	Used for binary files to specify the start of a field. The first byte of an input or output record is "1." Use `fieldStart` *or* `position` but not both.
`length`	Field length. Default is `1`.
`type`	Data type of this field. Valid values: `STRING`, `INTEGER`, `UNSIGNED INTEGER`, `SHORT`, `UNSIGNED SHORT`, `FLOAT`, `DOUBLE`, `BOOLEAN`, and `TIME`. The default is `STRING`.
`value`	Value of the field if it is a constant field. Otherwise, it contains the name of the message item that holds the data for this field.
`padCharacter`	Allows you to override the `padCharacter` specified in the `FILE_OPTIONS` section. If needed, specify an alphanumeric character.
`padDirection`	Allows you to override the `padDirection` specified in the `FILE_OPTIONS` section. If needed, specify either `left` or `right`.
`convertTo`	The TIBCO Rendezvous numeric data types `INTEGER`, `UNSIGNED INTEGER`, `SHORT`, `UNSIGNED SHORT`, and `FLOAT` can be converted to `PACKED`, `ZONED`, `BINARY`, or `Floating Point`. `STRING` values that are in numeric format can also be converted to `PACKED`, `ZONED`, `BINARY`, or `Floating Point` output.
`precision`	Specifies the size of the field and the number of decimals. This tag is not used for `COMP-1` or `COMP-2` fields.

When you specify `TIME` as a data type, FilePublisher gets the current system time, and places it in the TIBCO message. When FileSubscriber receives the message, it converts the time to a string, which represents the system time of the publishing system.

Tags in the constraint Parameter

FileSubscriber supports multiple line formats from different sections of a TIBCO message. To support this, you include multiple `FILE_LINE` elements in the FileType section. When multiple `FILE_LINE` elements are used, a `constraint` parameter must be supplied for each `FILE_LINE` element to indicate when to use that definition.

The FILE_LINE element's `constraint` parameter contains the following tags:

Tag	Description
`containerName`	The name of the TIBCO message that contains the fields that map to the `FILE_LINE` element.
`lineLength`	The length of a line in the output file for this line field definition. The default is the value specified by the `lineLength` parameter in the `FILE_OPTIONS` section.

Sample Configuration Elements

This section contains several configuration element examples that you can use as models.

Example 9   Record Mode Subscriber - Delimited Text File

This example shows the FileType section of a configuration file for processing TIBCO Rendezvous messages for a delimited file.

Configuration file name – `SUB001`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT1",
                 delimiter = "|",
                 skipPadding = "true",
                 isBinary = "false",
                 noOfRetries = "10", retryInterval = "5",
                 subscribeSubjectName = "SUB.PUB001.FT1",
                 autoGenerateFile = "true",
                 appendDateTime = "true",
                 saveFileInterval = "100",
                 transferType = "RecordMode"
}

FILE_LINE = {
field = { description = "Part Number", position = "1",
fromMessage = "true", value = "PartNo", type = "STRING" },
field = { description = "Part Description", position = "2",
fromMessage = "true", value = "Desc", type = "STRING" },
field = { description = "Manufacturer", position = "3",
fromMessage = "true", value = "Manufacturer", type = "STRING" },
field = { description = "Model", position = "4",
fromMessage = "true", value = "Model", type = "STRING" },
field = { description = "Quantity", position = "5",
fromMessage = "true", value = "Quantity", type = "INTEGER" },
field = { description = "Unit Price", position = "6",
fromMessage = "true", value = "Price", type = "FLOAT" },
field = { description = "Warranty Period in years", position = "7",
fromMessage = "true", value = "Warranty", type = "STRING" }
}
```

Assuming that the file used in the first FilePublisher example is being subscribed to, FileSubscriber would create a file that would look as follows:

```
115-01-0500|MONITOR|SONY|VIEWSONIC|1|350.500000|1 Year
115-15-6542|CPU-PIII750MHZ|COMPAQ|PRESARIO|1|900.000000|1 Year
115-67-7356|HDD20GB|SEAGATE|ST500|1|276.600006|1 Year
115-34-8767|FDD54|HP|T24333|1|86.000000|1 Year
115-77-5555|CDRW|HP|T75668|1|350.000000|1 Year
115-78-4646|KEYBOARD|COMPAQ|EASYKEY101|1|32.000000|1 Year
115-88-4454|MOUSE|MICROSOFT|M323|1|26.000000|1 Year
115-36-2727|WINDOWS2000|MICROSOFT|MSW2000|1|400.000000|1 Year
```

Example 10 Record Mode Subscriber - Fixed Length Text File

This example shows the FileType section of a configuration file for processing TIBCO Rendezvous messages for a fixed length text file.

Configuration file name – SUB002.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT2",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 useFixedRecordFile = "true",
                 appendToExistingFile = "false",
                 truncateRecords = "error",
                 exitOnFileSaveError = "false",
                 noOfRetries = "10", retryInterval = "5",
                 isCertified = "false",
                 subscribeSubjectName = "SUB.PUB002.FT2",
                 generateFileSubjectName = "ENDPUB.PUB002.FT2",
                 generateFileFieldName = "filename",
                 autoGenerateFile = "false",
                 transferType = "RecordMode"
}

FILE_LINE = {
field = { description = "Part Number", fromMessage = "true",
value = "PartNo", position="0", length = "11", type = "STRING" },
field = { description = "Part Description", fromMessage = "true",
value = "Desc", position="11", length = "15", type = "STRING" },
field = { description = "Vendor", fromMessage = "true",
value = "Vendor", position="26", length = "11", type = "STRING" },
field = { description = "Model", fromMessage = "true",
value = "Model", position="37", length = "12", type = "STRING" },
field = { description = "Quantity", fromMessage = "true",
value = "Quantity", position="49",length = "2", type = "INTEGER" },
field = { description = "Unit Price", fromMessage = "true",
value = "Price", position="51", length = "6", type = "FLOAT" },
field = { description = "Warranty Period in years",
fromMessage = "true", value = "Warranty", position="57",
length = "10", type = "STRING" }
}
```

Assuming that the file used in the second FilePublisher example is being subscribed to, FileSubscriber would create a file that looks as follows:

```
115-01-0500MONITOR SONY VIEWSONIC 2 250.501 Year
115-15-6542CPU-PIII750MHZ COMPAQ PRESARIO 4 12.3001 Year
115-67-7356HDD20GB SEAGATE ST500 2515.0001 Year
115-34-8767FDD54 HP T24333 1 45.2201 Year
115-77-5555CDRW HP T57213 5 290.001 Year
115-78-4646KEYBOARD COMPAQ EASYKEY1 10121.201 Year
```

Example 11 Record Mode Subscriber - Different Header and Detail Formats

In the FilePublisher example, a file with multiple record types was used as an example. This example shows how to configure the FileSubscriber to handle the TIBCO Rendezvous messages to process a multiple format file. The following steps could be used to set up the configuration file:

1. Use the FileType section to specify attributes of the file the subscriber creates.
2. Use a `FILE_LINE` element with a `containerName` parameter to define the format of each output line to be written after retrieving data from the message. FileSubscriber will create an output record for each `FILE_LINE` element.

   For the example shown, FileSubscriber will create a header record for the "Recfm1_Quote" container, then it will create three detail records from the "Recfm2_Items" record.
3. Specify the field attributes for the output record by using the `messageItem` parameter.
   - For fixed-length files, identify the field by using both the `position` and `length` parameters.
   - For delimited files, specify only a delimiter.

Configuration file name – `SUB003`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT3",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 useFixedRecordFile = "true",
                 appendToExistingFile = "false",
                 truncateRecords = "error",
                 exitOnFileSaveError = "false",
                 noOfRetries = "10", retryInterval = "5",
                 isCertified = "false",
                 subscribeSubjectName = "SUB.PUB003.FT3",
                 autoGenerateFile = "true",
                 saveFileInterval = "100",
                 transferType = "RecordMode"
}

FILE_LINE = {
constraint = { containerName = "Recfm1_Quote" },
field = { description = "Record Identifier", fromMessage = "false",
value = "QUOTE", position="0", length = "5", type = "STRING" },
field = { description = "Quote Request Number",
fromMessage = "true", value = "QuoteRequestNo", position="5",
length = "5", type = "INTEGER" },
field = { description = "Supplier Code", fromMessage = "true",
value = "SupplierCode", position="10",length = "5", type = "STRING"
}
FILE_LINE = {
constraint = { containerName = "Recfm2_Items" },
field = { description = "Record Identifier", fromMessage = "false",
value = "ITEMS", position="0",length = "5", type = "STRING" },
field = { description = "Part Number", fromMessage = "true",
value = "PartNo", position="5", length = "11", type = "STRING" },
field = { description = "Part Description", fromMessage = "true",
value = "Desc", position="16",length = "15", type = "STRING" },
field = { description = "Vendor", fromMessage = "true",
value = "Vendor", position="31",length = "11", type = "STRING" },
field = { description = "Model", fromMessage = "true",
value = "Model", position="42",length = "12", type = "STRING" },
field = { description = "Quantity", fromMessage = "true",
value = "Quantity", position="54",length = "2", type = "INTEGER" },
field = { description = "Unit Price", fromMessage = "true",
value = "Price", position="56",length = "6", type = "FLOAT" },
field = { description = "Warranty Period in years",
fromMessage = "true", value = "Warranty", position="62",
length = "10", type = "STRING" }
}
```

Assuming that the file used in the third FilePublisher example is being subscribed to, the file that FileSubscriber would create would look as follows:

```
QUOTE15004MC001
ITEMS115-01-0500MONITOR SONY VIEWSONIC 2 250.501 Year
ITEMS115-15-6542CPU-PIII750MHZ COMPAQ PRESARIO 4 12.3001 Year
ITEMS115-67-7356HDD20GB SEAGATE ST500 2515.0001 Year
QUOTE15004MC002
ITEMS115-34-8767FDD54 HP T24333 1 45.2201 Year
ITEMS115-77-5555CDRW HP T57213 5 290.001 Year
ITEMS115-78-4646KEYBOARD COMPAQ EASYKEY1 10121.201 Year
```

Example 12 Record Mode Subscriber - Supporting Numeric Data Types

This examples shows how a FileSubscriber configuration file can be coded to support the numeric data types.

Configuration file name – `SUB004`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="BIN2",
                 dataSetType = "SEQ",
                 isBinary = "true",
                 lineLength = "95",
                 useFixedRecordFile = "true",
                 appendToExistingFile = "false",
                 truncateRecords = "error",
                 exitOnFileSaveError = "false",
                 noOfRetries = "10",
                 retryInterval = "5",
                 isCertified = "false",
                 subscribeSubjectName = "SUB.PUB004.BIN2",
                 autoGenerateFile = "true",
                 saveFileInterval = "100",
                 transferType = "RecordMode"
}

FILE_LINE = {
field = { description = "Character", fromMessage = "true",
value = "Char", position="0", length="5", type="STRING" },
field = { description = "Packed", fromMessage = "true",
value = "Packed", position="5", type = "DOUBLE", precision="10,2",
convertTo="PACKED" },
field = { description = "zoned", fromMessage = "true",
value = "Zoned", position="11", type = "DOUBLE", precision="10,2",
convertTo="ZONED" },
field = { description = "binary", fromMessage = "true",
value = "Binary", position="21", type = "INTEGER",
precision="10,0", convertTo="BINARY" }
field = { description = "Float", fromMessage = "true",
value = "Float", position="29", type = "DOUBLE", precision="7,2",
convertTo="COMP-1" }
field = { description = "Double", fromMessage = "true",
value = "Double", position="33", type = "DOUBLE", precision="10,4",
convertTo="COMP-2" }
field = { description = "Hex", fromMessage = "true", value = "Hex",
position="41", length="10", type = "OPAQUE" }
field = { description = "Date", fromMessage = "true",
value = "Date", position="51", length="10", type = "OPAQUE" }
field = { description = "Time", fromMessage = "true",
value = "Time", position="61", length="8", type = "OPAQUE" }
field = { description = "STime", fromMessage = "true",
value = "STime", position="69", length="26", type = "OPAQUE" }
}
```

Example 13 Record Mode Subscriber with ECM

This example shows how to configure FileSubscriber to support ECM. In Record Mode ECM you need to specify `endPublishSubject` to signal the end of file publishing to the FileSubscriber. The `generateFileSubjectName` parameter in the FileSubscriber should be same as `endPublishSubject`.

Configuration file name – `SUB005`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT5",
                 delimiter = "|",
                 skipPadding = "true",
                 isBinary = "false",
                 noOfRetries = "10", retryInterval = "5",
                 subscribeSubjectName = "SUB.PUB005.FT1",
                 generateFileSubjectName = "ENDPUB.PUB005.FT1",
                 generateFileFieldName = "filename",
                 confirmationSubject = "ECM.PUB005.FT1",
                 autoGenerateFile = "false",
                 transferType = "RecordModeECM"
}

FILE_LINE = {
field = { description = "Part Number", position = "1",
fromMessage = "true", value = "PartNo", type = "STRING" },
field = { description = "Part Description", position = "2",
fromMessage = "true", value = "Desc", type = "STRING" },
field = { description = "Manufacturer", position = "3",
fromMessage = "true", value = "Manufacturer", type = "STRING" },
field = { description = "Model", position = "4",
fromMessage = "true", value = "Model", type = "STRING" },
field = { description = "Quantity", position = "5",
fromMessage = "true", value = "Quantity", type = "INTEGER" },
field = { description = "Unit Price", position = "6",
fromMessage = "true", value = "Price", type = "FLOAT" },
field = { description = "Warranty Period in years", position = "7",
fromMessage = "true", value = "Warranty", type = "STRING" }
}
```

Example 14 Block Mode Subscriber - Simple File Transfer

This configuration is for subscribing a file using Block Mode.

Configuration file name – `SUB006`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT6",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 useFixedRecordFile = "true",
                 appendToExistingFile = "false",
                 truncateRecords = "error",
                 exitOnFileSaveError = "false",
                 noOfRetries = "10", retryInterval = "5",
                 isCertified = "false",
                 subscribeSubjectName = "SUB.PUB006.FT2",
                 generateFileSubjectName = "ENDPUB.PUB006.FT2",
                 generateFileFieldName = "filename",
                 autoGenerateFile = "false",
                 transferType = "BlockModeSFT"
}
```

Example 15 Block Mode Subscriber with ECM

This configuration is for subscribing a file in Block Mode using ECM.

Configuration file name – `SUB007`.

The FileType section of the configuration file:

```
[FileType]
FILE_OPTIONS = { filePrefix="FT7",
                 dataSetType = "SEQ",
                 isBinary = "false",
                 lineLength = "100",
                 useFixedRecordFile = "true",
                 appendToExistingFile = "false",
                 truncateRecords = "error",
                 exitOnFileSaveError = "false",
                 noOfRetries = "10", retryInterval = "5",
                 isCertified = "false",
                 subscribeSubjectName = "SUB.PUB007.FT2",
                 useExplicitConfirmation = "true",
                 ECMSubscriberName = "ECM.PUB007.FT2",
                 transferType = "BlockModeECM"
}
```

Guaranteed Delivery for EMS Messages

This section describes the setup required for guaranteed delivery for EMS messages using durable subscribers.

Guaranteed delivery and receipt of EMS messages is enabled by sending persistent messages over topics to durable subscribers. For this purpose, only the transferTypes of `BlockModeSFT` and `RecordMode` with EMS are used.

Durable subscribers to a topic cause published messages to be saved while the durable subscriber is offline. Also, subscribers to a topic that are connected to a fault-tolerant server will receive messages from the secondary server after a failover

Follow these steps to achieve guaranteed delivery:

1. Set the `deliveryMode="persistent"` for the Publisher.

   Persistent messages published to a topic are written to disk only if that topic has at least one durable subscriber or one subscriber with a fault-tolerant connection to the EMS server.

2. Set the `publishDestinationType="TOPIC"` for the Publisher

   Similarly, for the subscriber, set the `subscribeDestinationName="TOPIC"`.

3. Set the `SubscriberName="sub1"`

   Here, `sub1` is an identifier of your choice.

   A new `SubscriberName` ID is required so that the EMS server can identify which consumer client(s) require(s) which messages resent after a failure. This Subscribername ID is created dynamically when the Subscriber is started. These durable subscriber Subscriber IDS can also be created with the aid of the EMS Administration Tool or through configuration files on the server. This first implementation of durable subscriber support in the Adapter does not have any mechanism to remove a durable subscriber through the Subscriber; the EMS Administration tool will have to be used.

4. Optional specify which EMS_ACK_TYPE acknowledgement for the Subscriber to use.

5. If the `endPublishDestinationName` parameter is specified in the Publisher, it must equal the same value as `publishDestinationName`. Similarly, the value of the `subscribeDestinationName` parameter and `publishDestinationName` must equal the value of the `generateFileSubjectName` parameter. in the Subscriber.

Chapter 4 Using the Adapter

This chapter contains a variety of topics for using TIBCO Adapter for Files (IBM i).

Topics

- Starting and Stopping the Adapter, page 98
- Block Transfer Mode, page 102
- File Transfer Using ECM, page 103
- Sending and Receiving Numeric Data Types, page 106
- Pre-Processing and Post-Processing Files, page 107
- Using Heartbeat Messages, page 109
- Working with Batch Messages, page 110
- Sending Trigger Messages, page 111
- FilePublisher Usage Guidelines, page 112
- FileSubscriber Usage Guidelines, page 115

Starting and Stopping the Adapter

This section describes how to start and stop the FilePublisher and FileSubscriber.

Starting FilePublisher

Before starting FilePublisher, ensure the TIBCO Rendezvous daemon is running. For example:

```
SBMJOB CMD(CALL PGM(TIBRV/RVD)) ALWMLTTHD(*YES)
```

If using a remote daemon, ensure that it is started with the `-permanent` option. Normally if the TIBCO Rendezvous daemon is not running, execution of FilePublisher starts the daemon automatically.

The Adapter, as well as the TIBCO Rendezvous daemon, require a multi-user threaded environment to run. The Command Entry interactive display does not provide this environment.

Using the SBMJOB Command

The FilePublisher program, `SXF3RPUB`, can be submitted to batch. Use the `SBMJOB` command to do this.

The following is an example of the `SBMJOB` command. Note that there are two parameters, `JOBMSGQFL` and `ALWMLTTHD`, that require a value other than the default.

```
ADDLIBLE TIBFALIB

SBMJOB CMD(call pgm(SXF3RPUB)
PARM('-config' 'TIBFASMPRV/INIFPUB(PUB005)'))
JOBMSGQFL(*PRTWRAP)
ALWMLTTHD(*YES)
```

Parameters in this `SBMJOB` command example were set as follows:

- The FilePublisher program, `SXF3RPUB`, is assumed to be in the library `TIBFALIB`.
- FilePublisher was submitted with the `-config` parameter. This causes FilePublisher to use configuration file member `PUB005` in file `INIFPUB`. If you omit this parameter, FilePublisher will use the sample configuration file, `TIBFALIB/INIFPUB(INIFPUB)`.
- The `JOBMSGQFL` parameter was set to `*PRTWRAP`, in case there should be a large number of messages issued to the job message queue.

- The ALWMLTTHD parameter must be set to *YES. This allows the job to run with multiple user threads.

Using a CL Program

One can fashion a CL program after the CL programs in TIBFASMPRV/QCLSRC to easily submit the FilePublisher to batch.

Using the QShell Environment

You can also start FilePublisher interactively in the QShell environment, as follows:

```
ADDLIBLE TIBFALIB
QSH
cd /usr/tibco/tibfa/TIBFALIB
sxf3rpub -config 'tibfalib/inifpub(pub003)'
```

or

```
sxf3rpub -service 7500 -daemon hawk400.na.tibco.com:7500
-config 'tibfalib/inifpub(pub003)'
```

Starting FileSubscriber

Before starting FileSubscriber, ensure the TIBCO Rendezvous daemon is running. For example:

```
SBMJOB CMD(CALL PGM(TIBRV/RVD)) ALWMLTTHD(*YES)
```

If using a remote daemon, ensure that it is started with the -permanent option. Normally if the TIBCO Rendezvous daemon is not running, execution of FileSubscriber starts the daemon automatically.

Using the SBMJOB Command

The FileSubscriber program, SXF3RSUB, can be submitted to batch. Use the SBMJOB command to do this.

The following is an example of the SBMJOB command. Note that there are two parameters, JOBMSGQFL and ALWMLTTHD, that require a value other than the default.

```
ADDLIBLE TIBFALIB

SBMJOB CMD(call pgm(SXF3RSUB)
PARM('-config' 'TIBFASMPRV/INIFSUB(SUB005)'))
JOBMSGQFL(*PRTWRAP)
ALWMLTTHD(*YES)
```

Parameters in this SBMJOB command example were set as follows:

- The FileSubscriber program, SXF3RSUB, is assumed to be in the library TIBFALIB.
- FileSubscriber was submitted with the -config parameter. This causes FileSubscriber to use the configuration file member SUB005 in file INIFSUB. If you omit the parm parameter, FileSubscriber will use the sample configuration file, TIBFALIB/INIFSUB(INIFSUB).
- The JOBMSGQFL parameter was set to *PRTWRAP, in case there should be a large number of messages issued to the job message queue.
- The ALWMLTTHD parameter must be set to *YES. This allows the job to run with multiple user threads.

Using a CL Program

One can fashion a CL program after the CL programs in TIBFASMPRV/QCLSRC to easily submit the FileSubscriber to batch.

Using the QShell Environment

You can also start FileSubscriber interactively in the QShell environment, as follows:

```
ADDLIBLE TIBFALIB
QSH
cd /usr/tibco/tibfa/TIBFALIB
sxf3rsub -config 'tibfalib/inifsub(sub003)'
```

or

```
sxf3rsub -service 7500 -daemon hawk400.na.tibco.com:7500
-config 'tibfalib/inifsub(sub003)'
```

Stopping FilePublisher

An active FilePublisher listens to the following terminate subject:

_FILEADAPTER.<*adapterName*>.TERMINATE

where <*adapterName*> is the name of the Adapter instance as configured in the FilePublisher configuration file. You can stop FilePublisher by sending a terminate message to this terminate subject.

Using the SBMJOB Command

You can use the following SBMJOB command to stop FilePublisher:

```
SBMJOB CMD(CALL PGM(TIBFALIB/SXFTRIGR) PARM('-service' 7500
'-daemon' 'hawk400.na.tibco.com:7500'
'FILEADAPTER.TEST_FILE_PUB.TERMINATE' 'any message'))
ALWMLTTHD(*YES)
```

Using the QShell Environment

You can also stop FilePublisher interactively in the QShell environment, as follows:

```
ADDLIBLE TIBFALIB
QSH
cd /usr/tibco/tibfa/TIBFALIB
sxftrigr -service 7500 -daemon hawk400.na.tibco.com:7500
_FILEADAPTER.TEST_FILE_PUB.TERMINATE  'any message'
```

Stopping FileSubscriber

An active FileSubscriber listens to the following terminate subject:

`_FILEADAPTER.`*<adapterName>*`.TERMINATE`

where *<adapterName>* is the name of the Adapter instance as configured in the FileSubscriber configuration file. You can stop FileSubscriber by sending a terminate message to this subject.

Using the SBMJOB Command

You can use the following `SBMJOB` command to stop FileSubscriber:

```
SBMJOB CMD(CALL PGM(TIBFALIB/SXFTRIGR) PARM('-service' 7500
'-daemon' 'hawk400.na.tibco.com:7500'
'FILEADAPTER.TEST_FILE_SUB.TERMINATE' 'any message'))
ALWMLTTHD(*YES)
```

Using the QShell Environment

You can also stop FileSubscriber interactively in the QShell environment, as follows:

```
ADDLIBLE TIBFALIB
QSH
cd /usr/tibco/tibfa/TIBFALIB
sxftrigr -service 7500 -daemon hawk400.na.tibco.com:7500
_FILEADAPTER.TEST_FILE_SUB.TERMINATE  'any message'
```

Block Transfer Mode

You may wish to publish a file without regard to its file structure. To do this, specify that the file is to be published, or subscribed to, in block mode. When a file is processed in block mode, the data from the file is read without regard to the field or record structure, and the data is published as a block.

Configuration

To enable block mode, specify the following parameters in the FileType section, `FILE_OPTIONS` element:

- `blockTransferMode` — When enabled , the file is published in blocks of data without further processing of the data (you must also specify the `endPublishSubject` parameter for FilePublisher and the `generateFileSubjectName` parameter for FileSubscriber.
- `blockTransferSize` – Specifies the block size, in bytes, that the Adapter uses to publish the file. The default value is 65536. This parameter is not used for FileSubscriber.

A file that is published in block transfer mode must be subscribed to in block transfer mode. Data received from a file that is published in block transfer mode does not have record or field characteristics. When the data is received by FileSubscriber, the blocks are written to a progress file. FileSubscriber generates a final output file only when it receives a message with the subject specified in `genFileSubjectName`. The `FILE_LINE` element is not required, and is ignored.

FileSubscriber receives a message with a subject that FileSubscriber has defined with the `generateFileSubjectName` parameter. When block transfer mode is enabled, the following tags are ignored for that file type in FileSubscriber:

- appendDateTime
- autoGenerateFile
- generateFileOnNumberOfMessages

When a file is published in block transfer mode, FilePublisher reads the file in blocks, as specified with the `blockTransferSize`, without additional processing of the data. The `MESSAGE_FIELD` element is not required, and is ignored.

File Transfer Using ECM

ECM is well suited for transferring files without having to define schemas. Both FilePublisher and FileSubscriber must be set up to use ECM for successful transfer.

In ECM, FilePublisher performs the following tasks:

- On initial startup, FilePublisher sends out an administration message to the ECM subscribers specified in its configuration file. This process is initiated only when FilePublisher starts for the first time.
- On restart, the active configuration information is retrieved from the progress file. Resynchronized handshakes are performed as needed to re-establish communication between the FilePublisher and its subscribers.
- When an inactive subscriber requests activation, FilePublisher marks the subscriber to be activated on a new file boundary.

In ECM, FileSubscriber performs the following tasks:

- On initial startup, FileSubscriber waits for an administration message from FilePublisher. Once the handshake has been established, FileSubscriber creates a progress file.

⚠ Do not edit or delete the progress file.

- When FileSubscriber is restarted, it uses the progress file it previously created and proceeds.
  - — If a file transfer was previously active, FileSubscriber continues from where it left off.
  - — If no file transfer is active, FileSubscriber waits for an administration message from FilePublisher and starts receiving file transfers on a new file boundary.

Error Handling

FilePublisher keeps track of the last block published in the progress file for restart purposes. When the Adapter encounters an error, it publishes a message using the error subject established in the configuration file (`ERROR_SUBJECT`).

FileSubscriber keeps track of the last block of data received and written in the progress file for restart purposes. When a subscriber detects an error (such as a locked file) in the middle of a file transfer, the subscriber converts the working file to an error file. When the Adapter encounters an error, it publishes a message using the error subject established in the configuration file (`ERROR_SUBJECT`).

TIBCO Rendezvous Subjects Used by ECM

The ECM administration messages are exchanged using the following subject names:

- `_TIBCO_AE_ADAPTER_FAFT_PUBTOSUB.`<*endpoint subject name*>
- `_TIBCO_AE_ADAPTER_FAFT_SUBTOPUB.`<*endpoint subject name*>

Administrators can listen to these subjects at runtime.

ECM Configuration

The following are used for FilePublisher configuration:

- `useExplicitConfirmation`: Flag that specifies whether Explicit Confirmation Mode is on or off during block transfer.

  All the options below are only valid if this flag is `true`. [default: “`false`”]
- `ECMSubsriberName`: This field specifies the corresponding ECM subscriber(s) name. When there are more than one participating subscribers, repeat this option and enter a unique subscriber name.
- `BlockTransferMode`: set to `true` to ensure block mode is used.
- `retransmissionDelayTicks`: Specifies the number of seconds the FileSubscriber should wait between retries to re-connect to the Publisher after a startup. It is used to coordinate the startup handshake used between the Publisher and Subscriber, when ECM is being used. Default is 10 seconds.
- `noWaitAfterConfirmations`: If `true`, the next transaction is processed immediately after FilePublisher receives all the confirmations. If `false`, it uses the normal `transactionDelay` parameter to process transactions. Default is "`true`".

The following are the FileSubscriber options.

- `useExplicitConfirmation`: Flag that specifies whether Explicit Confirmation Mode is on or off during block transfer. All the options below are only valid if this flag is true. [default: "`false`"]

- `ECMSubscriberName`: the ECM subscriber name. This entry must match the corresponding participating ECM Publisher.

TIBCO Adapter for Files (IBM i) and TIBCO Adapter for z/OS can communicate using the block mode and record mode using ECM only with Rendezvous.

TIBCO Adapter for Files (IBM i) and TIBCO Adapter for Files can communicate using ECM with Rendezvous only in the block mode.

Sending and Receiving Numeric Data Types

When sending records from FilePublisher with Packed Decimal fields (COMP-3), the message item description should set the type to either COMP-3 or PACKED. The packed data will be automatically converted to 8-byte floating point DOUBLE on the wire, unless the `convertToString` option is used. A convertToString="true" will cause the packed data to be converted in to a DISPLAY format, sent as STRING on the wire. This avoids any loss of precision and rounding that can occur if FLOAT is used to represent decimal numbers. Sample definitions are shown below:

Sent as DOUBLE:

```
messageItem = { position="4", precision="7,2",
                label="ITEM-PRICE", type="COMP-3" },
```

Sent as STRING:

```
messageItem = { position="4", precision="7,2",
                label="ITEM-PRICE", type="COMP-3",
                convertToString="true" },
```

When sending records with COMP-1 FLOAT, you may set the type field to COMP-1 or FLOAT. When sending COMP-2 DOUBLE, you may set the type field to COMP-2 or DOUBLE. When sending BINARY fields, you can either set the type field as SHORT (2 byte), INTEGER (4 byte) or LONGLONG (8 byte), or you may set the type field to BINARY or COMP, and explicitly set the length using the precision field as described above.

When receiving records at the Subscriber, Packed Decimal field (COMP-3) message descriptions should match what was sent from the Publisher. If the Publisher sent the data on the wire as DOUBLE, the type field should bet set to DOUBLE, and the convertTo option must be used, and set to COMP-3 or PACKED.   If the Publisher sent the data on the wire as STRING, the type field should bet set to STRING, and the convertTo option must be used, and set to COMP-3 or PACKED. Sample definitions are shown below:

Sent as DOUBLE:

```
field = { fieldStart="4", precision="7,2",
          value="ITEM-PRICE", type="DOUBLE",
          convertTo=COMP-3" },
```

Sent as STRING:

```
field = { fieldStart="4", precision="7,2",
          value="ITEM-PRICE", type="STRING",
          convertTo=COMP-3" },
```

Pre-Processing and Post-Processing Files

You can configure TIBCO Adapter for Files (IBM i) to perform the following pre- and post-processing tasks:

- Process a file before or after it is published, or before or after it is written.
- Execute an IBM i command or program before or after a file is published, or before or after an output file is generated.

To perform these actions, you use the `executeBeforeProcess` and `executeAfterProcess` configuration parameters; these parameters are found in the FileType section, `FILE_OPTIONS` element in the configuration file for either FilePublisher or FileSubscriber.

If you want to have a program called, you must include the IBM i command `call` as part of the parameter. If you specify a valid command or program name that is not in the FilePublisher's or FileSubscriber's library list, then you must include the library name in the parameter.

When a program is called by either FilePublisher or FileSubscriber, FilePublisher or FileSubscriber stops processing the current file until the called program has finished executing.

Pre-Processing Files

When you configure the Adapter to pre-process files, the following occurs:

- **FilePublisher** When FilePublisher has a file that is to be published, FilePublisher executes the command or program. You must provide for a parameter that FilePublisher will return to the command or program. When FilePublisher executes the command, it supplies the name of the file that is to be published as a parameter.
- **FileSubscriber** When FileSubscriber has an output file to be generated, FileSubscriber executes the command or program. You must provide for a parameter that FileSubscriber will pass to the command or call program. When FileSubscriber invokes the command, it passes the name of the file as a parameter.

Sample Configuration

Suppose that you wish to call a CL program named `PROCFILE`, which is in a library named `PGMLIB`, before you publish a file. In the FileType section, `FILE_OPTIONS` element of the FilePublisher configuration file, you would include the `executeBeforeProcess` parameter and include the file name, as follows:

```
executeBeforeProcess=”CALL PGMLIB/PROCFILE”
```

For the CL program, you must include a CL variable that will receive the name of the file that is to be published. For example:

```
PGM PARM(&FILENAME)
```

and

```
DCL   VAR(&FILENAME)  TYPE(*CHAR)  LEN(50)
```

When the CL program is called, the CL variable `&FILENAME` will contain the name of the library and the file that are to be published. For example, if the file to be published is `FT111`, and the library name of the output library is `TIBRVOUT`:

```
&FILENAME = ‘TIBRVOUT/FT111’
```

Post-Processing Files

When you configure the Adapter to post-process files, the following occurs:

- **FilePublisher** When FilePublisher has finished publishing a file, FilePublisher executes the command or program. You must provide for two parameters that FilePublisher will return to the command or program. When FilePublisher executes the command or program, it supplies the name of the file that was published and the file status as parameters. The file name parameter passed includes the library name. The file status parameter will be `0` if the file was published successfully, or `1` if there was an error publishing the file.
- **FileSubscriber** When FileSubscriber has finished generating an output file, FileSubscriber executes the command or program. You must provide for two parameters that FileSubscriber returns to the command or program. When FileSubscriber executes the command, it supplies the name of the file that was written and the file status as a parameter. The file name parameter passed includes the library name. The file status parameter will be “`0`” if the file was processed successfully, or “`1`” if there was an error processing the file.

Using Heartbeat Messages

Heartbeat messages provide an indication that FilePublisher or FileSubscriber is active. These messages can be monitored by TIBCO Hawk™. TIBOCO Hawk can send notifications or alerts when FilePublisher or FileSubscriber goes down.

Configuration

To enable heartbeat messages, specify the following elements in the Options section of the FilePublisher or FileSubscriber configuration file:

- `PUBLISH_HEARTBEAT` – To enable heartbeat messages, set this element to `true`. The default is `false`.
- `HEARTBEAT_TIME` – Specifies the interval between heartbeat messages. The default is `60000` milliseconds, or 60 seconds.

Message Format

The subject on which the Adapter publishes heartbeat messages is in the following format:

`_FILEADAPTER.`<*adaptername*>`.HEARTBEAT`

For FilePublisher, the format of the heartbeat message is FilePublisher start time and current time. For each configured file type, the format is:

- File prefix
- File extension (N/A for IBM i)
- File status — Idle or Busy (being published)
- If busy, number of messages (or blocks) published

For FileSubscriber, the format of the heartbeat message is FileSubscriber start time and current time. Then, for each file type, the format is file prefix and file extension (N/A for IBM i).

The Adapter does not provide TIBCO Hawk microagents. It only publishes Heartbeat messages at specified intervals as specified above. You can monitor key events that are processed by the Adapter by setting the `EEM_Subject` element and installing the TIBCO EEM monitoring software. For details, see BusinessEvents™ Messages on page 14.

Working with Batch Messages

If you want to subscribe to a batch of messages and generate the out file after receiving the entire batch, complete the following steps:

1. Publish the batch of records with a `publishSubjectName`, for example, `A.B`.
2. Using the subscriber, configure a subscription to the messages with the `subscribeSubjectName` set to `"A.B"` (in the FileSubscriber configuration file).
3. Set the `generateFileSubjectName="A.B"` in the FileSubscriber configuration file.
4. Set the `generateFileFieldName="STOP"` (can be any value, `STOP` is descriptive)
5. After publishing the entire batch, publish a message with `subjectName= "A.B"`, which contains the `STOP` field. For example.

   ```
    A.B
    {
     RVMSG_STRING   6   STOP   "12345"
    }
   ```

   or

   ```
   tibrvlisten: Listening to subject A.B
   [2002-01-22 20:30:14]: subject=A.B, message={STOP="TIBADAPTER"}
   ```

Note that this solution may not work in a multi-threaded environment. That is, if several FilePublishers are sending messages on the same subject, the above scenario will fail.

Sending Trigger Messages

Trigger messages can be sent with the file syntax shown below (QShell environment):

1. Start QShell:

   ```
   QSH
   ```

2. Change to the bin directory:

   ```
   CD /usr/tibco/tibfa/TIBFALIB
   ```

3. Send the following:

   ```
   sxftrigr -service 7500 -daemon hk-as400.na.tibco.com:7500
   TRIGGER.FB80.ONE  'TIBFALIB/FILE1'

   sxftrigr -service 7500 -daemon hk-as400.na.tibco.com:7500
   TRIGGER.FB80.ONE  'TIBFALIB/FILE1(MEMBERn)'

   sxftrigr -service 7500 -daemon hk-as400.na.tibco.com:7500
   TRIGGER.FB80.ONE  'TIBFALIB/FILE1(*FIRST)'

   sxftrigr -service 7500 -daemon hk-as400.na.tibco.com:7500
   TRIGGER.FB80.ONE  'TIBFALIB/FILE1(*LAST)'

   sxftrigr -service 7500 -daemon hk-as400.na.tibco.com:7500
   TRIGGER.FB80.ONE  'TIBFALIB/FILE1(*ALL)'
   ```

FilePublisher Usage Guidelines

This section provides some guidelines for using FilePublisher and explains special configuration options. In includes the following information:

- Sending Numeric Data Types on page 112
- Sending Data Untranslated (OPAQUE) on page 113
- Constructing a Subject Name from Data on page 113
- Publishing Double Values on page 114
- Pre-registering Subscribers on page 114

Sending Numeric Data Types

This section explains how FilePublisher can be configured to handle the numeric data types supported by the Adapter.

You can configure FilePublisher to publish non-text files by using the `isBinary=true` parameter and the `messageItem` parameter tags in the FileType section. When transmitted to a Subscriber, numeric data types are converted to `FLOAT`, `STRING` or `INTEGER` TIBCO data types.

When you configure one of these data types, you use the “`type`” tag to identify the type of numeric field. To specify a binary decimal field, you can specify `type=”binary”` or `type=”COMP”` or `type=”COMP-4”`. To specify packed decimal, you can specify `type=”PACKED”` or `type=”COMP-3”`.

Floating-point numeric fields are specified as either `COMP-1` or `COMP-2`. `COMP-1` indicates that the input field is a 4 byte long floating point number. `COMP-2` indicates that the input field is an 8 byte long floating point number.

Since there can be a loss of precision when converting to `FLOAT` from non-float values, it may be preferable to specify `STRING` as the published data type.

See Sample Configuration Elements on page 63 for an example of a FilePublisher configuration file that handles the supported numeric data types.

Sending Data Untranslated (OPAQUE)

You can configure the Adapter to publish a field without translating the data within the field. Usually, when FilePublisher reads a record from a file, it translates the data within each field to TIBCO Rendezvous "wire format," which is then published. When FileSubscriber receives the published message, it converts the "wire format" message to the appropriate data type for the platform that it is on.

If it is necessary to publish binary data other than the supported COBOL numeric data types, from one platform to another platform that supports the same data types (such as from one EBCDIC machine to another EBCDIC machine), you can specify `OPAQUE` for the data type. When you do this, the data in the field is published "as is," with no conversion. When the FileSubscriber receives the `OPAQUE` TIBCO Rendezvous message, it writes the data in that field to the field "as is."

To support zoned, binary, packed, floating-point, and opaque data types:

1. Set the `isBinary` parameter in the FILE_OPTIONS element to `true`.
2. Set the `type` tag in the `messageItem` parameter to the desired value.

Constructing a Subject Name from Data

Data from a file is normally published with a predefined subject name. However, if the subject name is not constant for all the lines of data in a file, and if the subject depends on a certain data field from the input file, specify a valid node within angle brackets (< >). A valid node is a label of a defined `messageItem` parameter prefixed by its container name separated by a period.

For example:

```
publishSubjectName = "CHECK.JAN.<ChKey.RoutingCode>.CLEARED"
```

uses data of the label `RoutingCode` in the container name `ChKey` to construct the `publishSubjectName`. That is, if the value of `RoutingCode` in the message to publish is 154363, the `publishSubjectName` for publishing, that message is `CHECK.JAN.154363.CLEARED`.

The configuration file coding necessary in this example would be as follows:

```
[FileType]
FILE_OPTIONS={…
PublishSubjectName="CHECK.JAN.<ChKey.RoutingCode>.CLEARED",
}
MESSAGE_FIELDS={
messageContainer={name="ChKey",
messageItem={position="0",length="6",
Label="RoutingCode", type="STRING"},
messageItem={ … },
messageItem={ … },
```

Constructing a Subject Name from Data is ONLY supported by Rendezvous. It is not supported for EMS.

Publishing Double Values

FilePublisher supports publishing of floating point numbers in three ways:

- Float (4 byte)
- Double (8 byte)
- STRING – can be used to avoid losing precision

The Adapter defaults to 6-digit precision for these values.

Pre-registering Subscribers

You can configure FilePublisher to specify a list of anticipated subscribers for the subjects it is publishing. By pre-registering the anticipated subscribers, the delivery of certified messages is guaranteed even if FileSubscribers start before or after the configured FilePublisher or even if a FileSubscriber is stopped and restarted later. To use this feature, both the FilePublisher and FileSubscriber should establish RVCM sessions with the `requireOldMessages` parameter set to `true`. To implement the pre-registering of subscribers, include this section in the FilePublisher configuration file:

```
[Pre-Register]
SUBSCRIBER = { listenerName = "<name of the subscribing session>",
               listeningSubject = "<name of the subject subscribed for>" },
SUBSCRIBER = { listenerName = "<name of the subscribing session>",
               listeningSubject = "<name of the subject subscribed for>" }
……
```

Currently, the maximum number of subscribers that can be pre-registered is 126.

FileSubscriber Usage Guidelines

This section discusses the following FileSubscriber usage guidelines:

- Receiving Numeric Data Types on page 115
- Adding Header and Trailer Records on page 115

Receiving Numeric Data Types

This section explains how FileSubscriber can be configured to handle the numeric data types supported by the Adapter.

FileSubscriber can be configured to create non-text files by using the `isBinary=true` parameter in the FileType section and two field parameter tags in the `FILE_LINE` element. This allows FileSubscriber to process non-text data types. The numeric data types are converted from incoming TIBCO data types of either `INTEGER`, `UNSIGNED INTEGER`, `SHORT`, `UNSIGNED SHORT`, `FLOAT`, or `STRING` into the desired target data type that is specified in the convertTo parameter. Because there can be a loss of precision when converting from `FLOAT` values, it may be preferable to specify `STRING` as the published data type. The numeric data type to be used for the output record for each field is specified by the tag `convertTo` in a field parameter of the `FILE_LINE` element.

To support zoned, binary, packed, and floating-point data types:

1. Set the `isBinary` parameter in the FILE_OPTIONS element to `true`.
2. Set the `convertTo` and `precision` tabs in the `FILE_LINE` element to desired values.

Since the fields of data type `BINARY(COMP, COMP-4)` are being published as `INTEGER`, mention the `type="INTEGER"` and `convertTo="BINARY"`.

Adding Header and Trailer Records

You can add a header record or a trailer record, or both, to an output file. The added record(s) can contain any text string and variable data.

To add a header record, use the `fileHeader` parameter in the `FILE_OPTIONS` element of the FileSubscriber configuration file. To add a trailer record, use the `fileTrailer` parameter in the `FILE_OPTIONS` element of the FileSubscriber configuration file. For each of these parameters, a text string and up to three variables — the number of records, a blank field, and the date and time — can be specified in the following format:

```
fileHeader=”<text>|[%<fieldWidth>,NUMBER_OF_RECORDS%]|
   %<fieldWidth>,BLANK%]|[<text>|%<fieldWidth>,DATE_TIME%]”

fileTrailer=”<text>|[%<fieldWidth>,NUMBER_OF_RECORDS%]|
   %<fieldWidth>,BLANK%]|[<text>|%<fieldWidth>,DATE_TIME%]”
```

where

<text> is any alphanumeric descriptive string. If included, it can appear anywhere within the quotation marks but must not be included within the square brackets ([ ]).

%<fieldWidth> is the number of characters in the related field; % is a C type substitution

NUMBER_OF_RECORDS is replaced with the number of records in the output file. If there are no errors, this number should correspond to the number of messages received to generate the file.

Because the Adapter receives records into a work file then generates the final output file, the NUMBER_OF_RECORDS variable can be used when specifying a header record.

DATE_TIME is the date and time in the format YYYYMMDDHHMMSS.

BLANK is a blank field that can be used to provide a space between the other variables.

FileHeader Example

Suppose you want to add a header record to an output file that shows the number of records received and the date and time that the output file was created, with two spaces between the fields. You would include the following parameters in the FILE_OPTIONS element of the FileSubscriber configuration file as follow:

```
FILE_OPTIONS = { prefix=”test”,
fileHeader=”[%4,NUMBER_OF_RECORDS%][%2,BLANK%][%14,DATE_TIME%]”,
subscribeSubjectName=…
```

Appendix A Trace Log Examples

This appendix shows trace logs that record a successful session and a failed session for FilePublisher and FileSubscriber. Information in these trace logs is produced according to trace levels specified by the configuration files for FilePublisher and FileSubscriber.

Topics

- Successful FilePublisher Session, page 134
- Failed FilePublisher Session, page 138
- Successful FileSubscriber Session, page 141
- Failed FileSubscriber Session, page 147

Successful FilePublisher Session

```
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0050I Copyright 2000,
2007 by TIBCO Software, Inc.
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0050I All Rights
Reserved.
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0051I
**************************************************************
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0052I TIBCO Adapter for
Files AS/400 start...
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0053I File Adapter
Publisher - version: 4.5.0 (2007-March-30)  GA
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0054I    build: Mar 22
2007
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0055I Using
configuration file: <TIBFASMPRV/INIFPUB(PUB001)>
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0355I Using trace file:
<TIBFASMPRV/PUBLOG(PUB001)>     Size=0
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0341I Trace Level: 2
TZ=N/A
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0368I Trace Option
Switches: NNNNNNNN
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_TRACE SXF0056I
**************************************************************
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_CFG_FL SXF0179I [===] OPTIONS
section:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_RV_PARM SXF0061I <--- RV_SESSION
Summary Begin
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_RV_PARM SXF0062I Name: FILE_PUB_001
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_RV_PARM SXF0063I Service:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_RV_PARM SXF0064I Network:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_RV_PARM SXF0065I Daemon: tcp:7500
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_RV_PARM SXF0066I ----> RV_SESSION
Summary End
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0163I Adapter Name:
FILE_PUB_001
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0160I Input Directory:
TIBFASMPRV
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0161I Process Directory:
FAPUBPROC
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0162I Output Directory:
FAPUBOUT
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0175I Continue on
Configuration Error: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0168I Max Concurrent
Jobs: 5
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0172I Print File
Options: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0177I Log to QSYSOPR
*MSGQ: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0352I EPM
Subject/Destination Name: <EPM_SUBJECT>
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0170I Error
Subject/Destination: ERROR.SUBJECT
```

```
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_OPT_AT SXF0173I Error Exit
Condition Code: 1
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_RVTRAN SXF0057I Initializing,
using RV client library version 7.5.2
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-INIT_RVTRAN SXF0058I Creating RV
session.  Hostcodepage=  Networkcodepage=
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_CFG_FL SXF0180I
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_CFG_FL SXF0181I [===] FILE TYPE
section:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0080I --> File Type
Options Begin: Num 0
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0081I File Prefix: FT1
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0083I File Extension:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0375I Transfer Type :
RecordMode
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0084I Data Set Type: SEQ
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0085I Use File Polling:
true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0086I Poll Interval
(milli-seconds): 5000
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0087I
Subject/Destination Name: SUB.PUB001.FT1
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0088I Start Publish
Subject/Destination:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0089I End Publish
Subject/Destination:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0090I Trigger
Subject/Destination Name:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0091I Trigger Field
Name:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0093I Input Directory:
TIBFASMPRV
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0094I Process Directory:
FAPUBPROC
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0095I Output Directory:
FAPUBOUT
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0096I Publish Start
Message: false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0097I Publish End
Message: false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0098I Execute Before
Process:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0099I Execute After
Process:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0100I Remove After
Process: alwaysDo
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0101I Messages Per
Transaction: 10
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0102I Transaction Delay
(milli-seconds): 2000
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0103I Block Transfer
Mode: false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0104I Block Transfer
Size: 65536
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0105I Use Explicit
Confirmation: false
```

```
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0106I Flex-mode ECM:
false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0112I Retransmission
Delay Tick Count: 10
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0113I No Wait After
Confirmations: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0114I Is Certified:
false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0115I RVCM Time Limit:
60
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0116I Binary input file:
false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0118I lineLength: 80
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0135I Delimiter: |
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0136I Start At Line: 1
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0137I Use Field Width:
false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0138I Remove Leading
Blanks: false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0139I Remove Trailing
Blanks: false
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0140I Use Sequential
Temp Files: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0141I Default (Polling)
User ID:
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOAD_FLOPT SXF0146I <-- File Type
Options End: Num 0
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOD_MSG_FP SXF0158I ---> Message
Fields Begin:   0 [0]
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_MSG_IPM SXF0154I Item: 0 - Label:
PartNo, Type: STRING, Value: , Multiple: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_MSG_IPM SXF0154I Item: 1 - Label:
Desc, Type: STRING, Value: , Multiple: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_MSG_IPM SXF0154I Item: 2 - Label:
Manufacturer, Type: STRING, Value: , Multiple: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_MSG_IPM SXF0154I Item: 3 - Label:
Model, Type: STRING, Value: , Multiple: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_MSG_IPM SXF0154I Item: 4 - Label:
Quantity, Type: INTEGER, Value: , Multiple: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_MSG_IPM SXF0154I Item: 5 - Label:
Price, Type: FLOAT, Value: , Multiple: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LD_MSG_IPM SXF0154I Item: 6 - Label:
Warranty, Type: STRING, Value: 1 Year, Multiple: true
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-LOD_MSG_FP SXF0159I <--- Message
Fields End:   0 [0]
2007 Mar 26 04:01:52 PBL INFO  [CFG] AD AE400-PRS_CFG_FL SXF0180I
2007 Mar 26 04:01:52 PBL INFO  [APP] AD AE400-COMP_Q_FILE SXF1000I Job Queue
File=<FAPUBPROC/fpqueue>.
2007 Mar 26 04:01:52 PBL INFO  [APP] AD AE400-FAPUB_MAIN SXF1519I Setting up
terminate adapter Subject/Destination <_FILEADAPTER.FILE_PUB_001.TERMINATE>
2007 Mar 26 04:01:52 PBL INFO  [APP] AD AE400-INT_FL_PLR SXF1060I <FT1,> File
Polling requested.  configValid=1  admin_SubCount=-1042128396
2007 Mar 26 04:01:52 PBL INFO  [APP] AD AE400-FAPUB_MAIN SXF1525I Completed
Initialization and any Recovery of Publisher.
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-SCH_JOBQ SXF1009I <FT1> Publishing
file: <FT11> with UserId=
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-PB_EPM_ADV SXF2324I <FT11> Publishing
EPM Advisory message using TrackingId= from Subject/Destination=SUB.PUB001.FT1 with
AdvisoryType=10
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-PUBLS_FILE SXF2060I <FT1> file=<FT11>
recov=0  Tracking ID=Thk@@C@@FUScGk@@WVzzwDn@zzw
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-MOVE_FILE SXF2002I Moving file: <FT11>
from <TIBFASMPRV> to <FAPUBPROC>
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-MOVE_FILE SXF2003I Move file <FT11> to
directory <FAPUBPROC> completed successfully, and previous output
file=<FAPUBPROC/FT11> was deleted.
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-PUBLS_FILE SXF2057I <FT1> SEQ/GDG/PDS
file=<FT11> DSName=<FAPUBPROC/FT11>. Dynamic recl=80  bufsize=160  vb=1
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-PB_EPM_MSG SXF2092I <FT1> <FT11>
Publishing status message using TrackingId=Thk@@C@@FUScGk@@WVzzwDn@zzw from
Subject/Destination=SUB.PUB001.FT1 and JCL CC=0. epm=1
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-PUBLS_FILE SXF2059I <FT1> Start
publishing using PROGRESS file=<FAPUBPROC/PRGFT1>.
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-PUBLS_FILE SXF2064I <FT1> Start
publishing RECORD MODE file=<FT11> tranDelay timer=4.243992E-311,  ECM=-588860155
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-HDL_EOF_RECORD SXF2334I <FT11>
handleEOF RecordMode invoked. msgcont=0 eofsm=0 ewait=0 lrec=8 remove=1 haserr=0
remerr=0 execc=0 execcmd=<>
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-PB_EPM_MSG SXF2092I <FT1> <FT11>
Publishing status message using TrackingId=Thk@@C@@FUScGk@@WVzzwDn@zzw from
Subject/Destination=SUB.PUB001.FT1 and JCL CC=0. epm=1
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-HDL_EOF_RECORD SXF2079I <FT1> Finished
publishing RECORD MODE file: <FT11>
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-HDL_EOF_RECORD SXF2082I Removing file:
<FAPUBPROC/FT11>
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-HDL_EOF_RECORD SXF2083I Completed
processing file: <FT11>, no errors
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-DN_PUB_CLB SXF1014I <FT1> File Done
Callback. Completed file: <FT11>  using ECM=0.
2007 Mar 26 04:02:35 PBL INFO  [APP] AD AE400-FIN_JOB SXF1001I FinishJob status:
NumConCurrJobs=0   MaxConcur=5   NumQueued=0   JobQ=0
```

Failed FilePublisher Session

```
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0050I Copyright 2000,
2007 by TIBCO Software, Inc.
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0050I All Rights
Reserved.
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0051I
**************************************************************
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0052I TIBCO Adapter for
Files AS/400 start...
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0053I File Adapter
Publisher - version: 4.5.0 (2007-March-30)  GA
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0054I    build: Mar 22
2007
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0055I Using
configuration file: <fa45samp/inifpub(pub001)>
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0355I Using trace file:
<TIBFASMPRV/PUBLOG(PUB001)>     Size=0
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0341I Trace Level: 2
TZ=N/A
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0368I Trace Option
Switches: NNNNNNNN
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_TRACE SXF0056I
**************************************************************
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_CFG_FL SXF0179I [===] OPTIONS
section:
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-LD_RV_PARM SXF0061I <--- RV_SESSION
Summary Begin
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-LD_RV_PARM SXF0062I Name:
FILE_PUB_001
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-LD_RV_PARM SXF0063I Service:
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-LD_RV_PARM SXF0064I Network:
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-LD_RV_PARM SXF0065I Daemon: tcp:7500
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-LD_RV_PARM SXF0066I ----> RV_SESSION
Summary End
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0160I Input Directory:
TIBFASMPRV
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0161I Process
Directory: FAPUBPROC
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0162I Output Directory:
FAPUBOUT
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0175I Continue on
Configuration Error: true
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0168I Max Concurrent
Jobs: 5
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0172I Print File
Options: true
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0177I Log to QSYSOPR
*MSGQ: true
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0352I EPM
Subject/Destination Name: <EPM_SUBJECT>
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0170I Error
Subject/Destination: ERROR.SUBJECT
```

```
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-PRS_OPT_AT SXF0173I Error Exit
Condition Code: 1
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_RVTRAN SXF0057I Initializing,
using RV client library version 7.5.2
2007 Mar 26 12:19:20 PBL INFO  [CFG] 131 AE400-INIT_RVTRAN SXF0058I Creating RV
session.  Hostcodepage=  Networkcodepage=
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-PRS_CFG_FL SXF0180I
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-PRS_CFG_FL SXF0181I [===] FILE TYPE
section:
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0080I --> File Type
Options Begin: Num 0
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0081I File Prefix: FT1
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0083I File Extension:
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0375I Transfer Type :
RecordMode
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0084I Data Set Type:
SEQ
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0085I Use File Polling:
true
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0086I Poll Interval
(milli-seconds): 5000
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0087I
Subject/Destination Name: SUB.PUB001.FT1
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0088I Start Publish
Subject/Destination:
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0089I End Publish
Subject/Destination:
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0090I Trigger
Subject/Destination Name:
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0091I Trigger Field
Name:
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0093I Input Directory:
TIBFASMPRV
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0094I Process
Directory: FAPUBPROC
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0095I Output Directory:
FAPUBOUT
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0096I Publish Start
Message: false
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0097I Publish End
Message: false
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0098I Execute Before
Process:
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0099I Execute After
Process:
2007 Mar 26 12:19:25 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0100I Remove After
Process: runJCL
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0101I Messages Per
Transaction: 10
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0102I Transaction Delay
(milli-seconds): 2000
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0103I Block Transfer
Mode: false
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0104I Block Transfer
Size: 65536
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0105I Use Explicit
Confirmation: false
```

```
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0106I Flex-mode ECM:
false
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0112I Retransmission
Delay Tick Count: 10
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0113I No Wait After
Confirmations: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0114I Is Certified:
false
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0115I RVCM Time Limit:
60
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0116I Binary input
file: false
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0118I lineLength: 80
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0135I Delimiter: |
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0136I Start At Line: 1
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0137I Use Field Width:
false
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0138I Remove Leading
Blanks: false
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0139I Remove Trailing
Blanks: false
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0140I Use Sequential
Temp Files: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0141I Default (Polling)
User ID:
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOAD_FLOPT SXF0146I <-- File Type
Options End: Num 0
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOD_MSG_FP SXF0158I ---> Message
Fields Begin:   0 [0]
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LD_MSG_IPM SXF0154I Item: 0 - Label:
PartNo, Type: STRING, Value: , Multiple: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LD_MSG_IPM SXF0154I Item: 1 - Label:
Desc, Type: STRING, Value: , Multiple: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LD_MSG_IPM SXF0154I Item: 2 - Label:
Manufacturer, Type: STRING, Value: , Multiple: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LD_MSG_IPM SXF0154I Item: 3 - Label:
Model, Type: STRING, Value: , Multiple: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LD_MSG_IPM SXF0154I Item: 4 - Label:
Quantity, Type: INTEGER, Value: , Multiple: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LD_MSG_IPM SXF0154I Item: 5 - Label:
Price, Type: FLOAT, Value: , Multiple: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LD_MSG_IPM SXF0154I Item: 6 - Label:
Warranty, Type: STRING, Value: 1 Year, Multiple: true
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-LOD_MSG_FP SXF0159I <--- Message
Fields End:   0 [0]
2007 Mar 26 12:19:26 PBL INFO  [CFG] 131 AE400-PRS_CFG_FL SXF0180I
SXF0325E Pub PRS_CFG_FL: Missing ADAPTER_NAME definition
```

Successful FileSubscriber Session

```
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5050I Copyright 2000,
2007 by TIBCO Software, Inc.
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5050I All Rights
Reserved.
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5051I
**************************************************************
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5052I TIBCO Adapter for
Files AS/400 start...
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5053I File Adapter
Subscriber - version: 4.5.0 (2007-March-30)  GA
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5054I    build: Mar 22
2007
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5055I Using
configuration file: <TIBFASMPRV/INIFSUB(SUB001)>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5322I Using trace file:
<TIBFASMPRV/SUBLOG(SUB001)>     Size=2000000
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5305I Trace Level: 2
TZ=N/A
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF0368I Trace Option
Switches: NNNNNNNN
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_TRACE SXF5056I
**************************************************************
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_CFG_FL SXF5186I
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_CFG_FL SXF5187I [===] OPTIONS
section:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LOAD_RVSPM SXF5061I <--- RV_SESSION
Summary Begin
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LOAD_RVSPM SXF5062I Name:
FILE_SUB_001
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LOAD_RVSPM SXF5063I Service:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LOAD_RVSPM SXF5064I Network:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LOAD_RVSPM SXF5065I Daemon: tcp:7500
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LOAD_RVSPM SXF5066I ----> RV_SESSION
Summary End
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5171I Adapter Name:
FILE_SUB_001
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5170I Output Directory:
FASUBOUT
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5182I Continue on
Configuration Error: true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5179I Print File
Options: true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5184I Terminate on RV
Send Error: true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5185I Log to QSYSOPR
*MSGQ: true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5321I EPM
Subject/Destination Name: <EPM_SUBJECT>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5177I Error
Subject/Destination: ERROR.SUBJECT
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_OPT_AT SXF5180I Error Exit
Condition Code: 1
```

```
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_RVTRAN SXF0057I Initializing,
using RV client library version 7.5.2
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-INIT_RVTRAN SXF0058I Creating RV
session.  Hostcodepage=  Networkcodepage=
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_CFG_FL SXF5186I
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_CFG_FL SXF5188I [===] FILE TYPE
section:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5080I --> File Type
Options Begin: Num 0
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5081I File Prefix: FT1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5083I File Extension:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5327I Transfer Type :
RecordMode
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5084I Data Set Type:
SEQ
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5085I
Subject/Destination Name: SUB.PUB001.FT1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5087I Generate File
Publish Subject/Destination:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5088I Generate File On
Number of Messages: 0
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5089I Publish Generate
File Message: false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5090I Save File
Interval (milli-seconds): 100
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5091I Auto Generate
File: true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5092I Generate File
Field Name: FileName
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5093I Process
Directory:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5095I Output Directory:
FASUBOUT
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5096I Append Date/Time:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5097I Append File
sequence number: false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5098I Force published
filename: false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5099I Execute Before
Process:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5100I Execute After
Process:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5101I Block Transfer
Mode: false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5102I Block Transfer
Size: 65536
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5103I Use Explicit
Confirmation: false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5104I Confirmation
Subject/Destination:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5109I Is Certified:
false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5110I Binary output
file: false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5111I Use Fixed Record
File: true
```

```
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5113I lineLength: 80
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5115I Primary Space
Allocation: 2
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5116I Secondary Space
Allocation: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5118I Append To An
Existing File: false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5119I Exit On File Save
Error: false
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5120I Max number of
retries for a locked Target file: 10
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5121I Retry Interval
(seconds): 5
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5142I Delimiter: |
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5143I Pad Character: '
'
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5144I Pad Direction:
right
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5145I Skip Padding:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5146I File Header:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5147I File Trailer:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FOPT_PR SXF5153I <-- File Type
Options End: Num 0
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5154I -----> Begin Line
Fields: 0 [0]
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 0
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5157I Description: Part
Number
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5160I Position: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5164I Value: <PartNo>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 0
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5157I Description: Part
Description
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5160I Position: 2
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5164I Value: <Desc>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
```

```
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 2
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5157I Description:
Manufacturer
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5160I Position: 3
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5164I Value:
<Manufacturer>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 2
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 3
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5157I Description:
Model
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5160I Position: 4
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5164I Value: <Model>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 3
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 4
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5157I Description:
Quantity
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5160I Position: 5
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5163I Type: INTEGER
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5164I Value: <Quantity>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 4
```

```
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 5
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5157I Description: Unit
Price
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5160I Position: 6
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5163I Type: FLOAT
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5164I Value: <Price>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 5
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 6
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5157I Description:
Warranty Period in years
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5160I Position: 7
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5164I Value: <Warranty>
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 6
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-LD_FLN_PRM SXF5168I <----- Line
Fields: 0 [0]
2007 Mar 26 04:01:52 SUB INFO  [CFG] 1BB AE400-PRS_CFG_FL SXF5186I
2007 Mar 26 04:01:52 SUB INFO  [APP] 1BB AE400-FASUB_MAIN SXF6011I Setting up
terminate adapter Subject/Destination <_FILEADAPTER.FILE_SUB_001.TERMINATE>
2007 Mar 26 04:01:52 SUB INFO  [APP] 1BB AE400-READ_CUR_STERR SXF7373I
<FASUBOUT/WKFT1> STERR num file=FASUBOUT/STERRFT1 does not exist for Prefix=FT1.
Will create it if needed.
2007 Mar 26 04:01:52 SUB INFO  [APP] 1BB AE400-SET_SUBSCRS SXF7113I <SUB.PUB001.FT1>
Setting up subscribe to Subject/Destination for FileType Prefix=FT1
2007 Mar 26 04:01:52 SUB INFO  [APP] 1BB AE400-FASUB_MAIN SXF6016I Completed
Initialization and any Recovery of Subscriber.
2007 Mar 26 04:02:35 SUB INFO  [APP] 1BB AE400-OPN_FILES SXF7009I <FASUBOUT/WKFT1>
TEXT file lineLng=80  recl=80  blksize=320. Full name=NT_fldata_returned_filename
2007 Mar 26 04:03:32 SUB INFO  [APP] 1BB AE400-GEN_OUT_FL SXF7020I <FASUBOUT/WKFT1>
Generating file for FileType=FT1 - for criterion=auto generate timer. nl=8  Blk=0.
errflags=0:0  Caller=saveTimerCallback
2007 Mar 26 04:03:32 SUB INFO  [APP] 1BB AE400-GEN_OUT_FL SXF7028I <FT1>
<FASUBOUT/WKFT1> Closed working file
2007 Mar 26 04:03:32 SUB INFO  [APP] 1BB AE400-GEN_OUT_FL SXF7038I <FT1>
<FASUBOUT/WKFT1> Generating target file=<FASUBOUT/FT1040332> 8 lines, 8 messages,
no errors
```

```
2007 Mar 26 04:03:32 SUB INFO  [APP] 1BB AE400-GEN_OUT_FL SXF7044I <FT1>
appendToExisting=False, so Removed PREV output file=<FASUBOUT/FT1040332>
2007 Mar 26 04:03:32 SUB INFO  [APP] 1BB AE400-GEN_OUT_PP SXF7053I <FASUBOUT/PRGFT1>
Progress file has been closed. EOF reason=auto generate timer
2007 Mar 26 04:03:32 SUB INFO  [APP] 1BB AE400-GEN_OUT_PP SXF7317W <FT1> FileType -
Deleting PRG file
```

Failed FileSubscriber Session

```
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5050I Copyright 2000,
2007 by TIBCO Software, Inc.
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5050I All Rights
Reserved.
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5051I
**************************************************************
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5052I TIBCO Adapter for
Files AS/400 start...
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5053I File Adapter
Subscriber - version: 4.5.0 (2007-March-30)  GA
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5054I    build: Mar 22
2007
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5055I Using
configuration file: <fa45samp/inifsub(sub001)>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5322I Using trace file:
<TIBFASMPRV/SUBLOG(SUB001)>     Size=2000000
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5305I Trace Level: 2
TZ=N/A
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF0368I Trace Option
Switches: NNNNNNNN
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_TRACE SXF5056I
**************************************************************
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_CFG_FL SXF5186I
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_CFG_FL SXF5187I [===] OPTIONS
section:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LOAD_RVSPM SXF5061I <--- RV_SESSION
Summary Begin
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LOAD_RVSPM SXF5062I Name:
FILE_SUB_001
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LOAD_RVSPM SXF5063I Service:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LOAD_RVSPM SXF5064I Network:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LOAD_RVSPM SXF5065I Daemon: tcp:7500
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LOAD_RVSPM SXF5066I ----> RV_SESSION
Summary End
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_OPT_AT SXF5170I Output Directory:
FASUBOUT
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_OPT_AT SXF5182I Continue on
Configuration Error: true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_OPT_AT SXF5179I Print File
Options: true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_OPT_AT SXF5184I Terminate on RV
Send Error: true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_OPT_AT SXF5185I Log to QSYSOPR
*MSGQ: true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_OPT_AT SXF5321I EPM
Subject/Destination Name: <EPM_SUBJECT>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_OPT_AT SXF5177I Error
Subject/Destination: ERROR.SUBJECT
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_OPT_AT SXF5180I Error Exit
Condition Code: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_RVTRAN SXF0057I Initializing,
using RV client library version 7.5.2
```

```
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-INIT_RVTRAN SXF0058I Creating RV
session.  Hostcodepage=  Networkcodepage=
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_CFG_FL SXF5186I
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_CFG_FL SXF5188I [===] FILE TYPE
section:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5080I --> File Type
Options Begin: Num 0
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5081I File Prefix: FT1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5083I File Extension:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5327I Transfer Type :
RecordMode
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5084I Data Set Type:
SEQ
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5085I
Subject/Destination Name: SUB.PUB001.FT1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5087I Generate File
Publish Subject/Destination:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5088I Generate File On
Number of Messages: 0
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5089I Publish Generate
File Message: false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5090I Save File
Interval (milli-seconds): 100
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5091I Auto Generate
File: true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5092I Generate File
Field Name: FileName
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5093I Process
Directory:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5095I Output Directory:
FASUBOUT
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5096I Append Date/Time:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5097I Append File
sequence number: false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5098I Force published
filename: false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5099I Execute Before
Process:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5100I Execute After
Process:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5101I Block Transfer
Mode: false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5102I Block Transfer
Size: 65536
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5103I Use Explicit
Confirmation: false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5104I Confirmation
Subject/Destination:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5109I Is Certified:
false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5110I Binary output
file: false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5111I Use Fixed Record
File: true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5113I lineLength: 80
```

```
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5115I Primary Space
Allocation: 2
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5116I Secondary Space
Allocation: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5118I Append To An
Existing File: false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5119I Exit On File Save
Error: false
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5120I Max number of
retries for a locked Target file: 10
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5121I Retry Interval
(seconds): 5
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5142I Delimiter: |
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5143I Pad Character: '
'
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5144I Pad Direction:
right
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5145I Skip Padding:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5146I File Header:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5147I File Trailer:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FOPT_PR SXF5153I <-- File Type
Options End: Num 0
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5154I -----> Begin Line
Fields: 0 [0]
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 0
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5157I Description: Part
Number
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5160I Position: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5164I Value: <PartNo>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 0
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5157I Description: Part
Description
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5160I Position: 2
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5164I Value: <Desc>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
```

```
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 2
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5157I Description:
Manufacturer
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5160I Position: 3
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5164I Value:
<Manufacturer>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 2
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 3
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5157I Description:
Model
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5160I Position: 4
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5164I Value: <Model>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 3
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 4
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5157I Description:
Quantity
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5160I Position: 5
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5163I Type: INTEGER
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5164I Value: <Quantity>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 4
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 5
```

```
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5157I Description: Unit
Price
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5160I Position: 6
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5163I Type: FLOAT
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5164I Value: <Price>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 5
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5156I --> Begin Line
Field: 6
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5157I Description:
Warranty Period in years
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5158I From Message:
true
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5160I Position: 7
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5161I Length: 1
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5162I Precision:
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5163I Type: STRING
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5164I Value: <Warranty>
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5165I Pad Character: '
'
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5166I Pad Direction:
right
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5167I <-- End Line
Field: 6
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-LD_FLN_PRM SXF5168I <----- Line
Fields: 0 [0]
2007 Mar 26 12:19:47 SUB INFO  [CFG] 136 AE400-PRS_CFG_FL SXF5186I
SXF5287E Sub PRS_CFG_FL: Missing ADAPTER_NAME definition
```

Appendix B Error Messages

This appendix describes the error messages used by the Adapter.

Topics

- Error Message Format, page 154
- Publisher Error Messages, page 155
- Subscriber Error Messages, page 201

Error Message Format

Messages that can appear in trace logs contain several repetitive preamble fields that you can ignore. These preamble fields are omitted from the messages in this appendix. For example, a complete message in a trace log is shown in this appendix as follows:

```
2004 March 5 19:24:03 PBL ERROR [CFG] SXF7145E HDL_TEXT_ERR <%s> reached...
```

Message in this appendix
starts here

The Adapter produces several categories of messages. The last letter of the error number, I, E, or W, indicates the type of message.

- Informational (INFO) messages confirm that an event, task, operation, or condition is normal. No action is necessary; therefore, INFO messages are omitted from this appendix.
- Warning messages indicate that there is a problem but that processing will continue.
- Error messages indicate a problem so severe that processing cannot continue for that file type.

A `%` symbol indicates a C type substitution. For example, `%s` indicates that the error message includes a string value as a substitute for the `%s` value.

Publisher Error Messages

SXF0000 – SXF0500

Number	Message	Description
SXF0192W	<Delimiter> can not be used when useFieldWidth is true	You have specified both a delimiter character and useFieldWidth=true. Use only one of the parameters for each file type.
SXF0193W	Invalid messagesPerTransaction value <%d>	The parameter was specified with an invalid value. Correct the value. Ensure that the value is greater than zero.
SXF0194W	messagesPerTransaction set to default value: <%d>	No value was specified in the configuration file for this parameter. Add this parameter if the default value is not the value you wish to be used.
SXF0195W	Not a Certified Session. Cannot pre-register listener %s for subject %s	You specified pre-registration, but the session is not a certified session. Either specify a certified session, or omit the pre-registration entries.
SXF0196W	File not found: %s	The file indicated was not found. Verify that the file exists and is available to FilePublisher.
SXF0197W	Library not found: %s	The library indicated was not found. Verify that the file exists and is available to FilePublisher.
SXF0198W	Directory not found: %s	The specified directory was not found. Make sure that the specified directory exists and is available.

Number	Message	Description
SXF0199E	Config parse error near line %d, failed to parse value	The format or value for a name/tag is not correct. A supplied value is incomplete; typically the value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure all values in the configuration file have matching quotes and braces, and that there are equal signs where required. Check the name/value tags in the section defined in the configuration file. Refer to the manual for the proper format.
SXF0204E	Max record COUNT must be > 0	A Max record count of zero is invalid. It says you want zero records from the file, which is meaningless.
SXF0207E	Duplicate retransmissionDelay entry. Ignored !	The retransmissionDelay was previously specified. A duplicate entry was encountered, and will be ignored.
SXF0208E	Invalid value set for %s	The %s contains the name of the FileAdapter parameter that has been set to an invalid value in the FileAdapter INI file.
SXF0209E	Unable to allocate memory for ECMSubscriber entry	An out of memory error was reported.
SXF0210E	Config error: ECMSubscriber defined without useExplicitConfirmation=true. ECMSubscriber ignored.	A mis-match in the configuration has been configured. Either also turn on the useExplicitConfirmation=true parameter, or eliminate the ECMSubscriber definition.
SXF0211E	Unknown/Invalid configuration parameter <%s> at line %d. Ignored	The %s contains the name of the FileAdapter parameter that has been set to an invalid value in the FileAdapter INI file, and the Line number indicates which FileType or other definition contains the parameter in error.
SXF0212E	Invalid file type: prefix more than 8 char	Names used in File type PREFIX= definitions are limited to 8 bytes in length.
SXF0213E	Unsupported dataSetType <%s>	The dataSetType= parameter can only be SEQ.

Number	Message	Description
SXF0217E	Missing parameter <%s>	The %s contains the name of the FileAdapter parameter that is missing from the FileAdapter INI file.
SXF0218E	Config ERROR: you have defined Record ECM and Block Mode ECM in the same Filetype. Config has been overriden to use Block Mode ECM.	You cannot use both Block mode and Record mode in the same FileType definition. Make the definition use one mode or the other.
SXF0219E	File cannot be defined as BINARY yet have useFieldWidth=%s	Setting a file type to BINARY is mutually exclusive with useFieldWidth=No. Change one or the other parameter.
SXF0220E	Input library [%s] cannot be the same as the process library.	The input library and the process library cannot be the same. Specify unique input and process libraries.
SXF0221E	Process library [%s] cannot be the same as the output library.	The output library and the process library cannot be the same. Specify unique output and process libraries.
SXF0222E	Invalid input library name: <%s>	The specified input library name is invalid. Use a valid library name. Check for invalid characters. Name length is 10 characters maximum.
SXF0223E	Invalid output library name: <%s>	The specified library name is invalid. Use a valid library name. Check for invalid characters. Name length is 10 characters maximum.
SXF0224E	Invalid process library name: <%s>	The library name specified is invalid. Use a valid library name. Check for invalid characters. Name length is 10 characters maximum.
SXF0225E	Pre-Register Sections appears before Options section near line %d	The Pre-Register section is specified in the configuration file in an invalid location. Specify the Pre-Register section after the Options section.

Number	Message	Description
SXF0226E	Duplicate definition for [%s] section	The section specified appears more than once in the configuration file. Correct the configuration file.
SXF0227E	Config parse error near line %d, invalid section name: '%s'	A section name in the configuration file is incorrect. A section name must be one of Trace, Options, Pre-Register, or FileType. Check the spelling of the section names of the configuration file. Names are case sensitive.
SXF0228E	Config parse error near line %d, failed to find ']'	In the configuration file, a section description, Trace, Options, PreRegister, or FileType, does not have a closing ']'. Check the configuration file and make sure that all section descriptions are formed correctly.
SXF0229E	Config parse error near line %d, failed to find section	In the configuration file, a section description is not one of Trace, Option, Pre-Register, or FileType. Check the configuration file, and ensure that the section descriptions are valid. Descriptions are case sensitive.
SXF0230E	Config parse error near line %d, failed to end expression	The configuration file does not have matching opening and closing '}'. Check the configuration file, and make sure that there are matching opening and closing '}'.
SXF0231E	Missing [%s] section	The configuration file is missing the specified section. You must specify the indicated section. Add the section to the configuration file.
SXF0232E	Missing [%s] section	The configuration file is missing the specified section. You must specify the indicated section. Add the section to the configuration file.
SXF0233E	Missing [%s] section	The configuration file is missing the specified section. You must specify the indicated section. Add the section to the configuration file.

Number	Message	Description
SXF0234E	Missing [%s] section option: %s	The configuration file is missing the specified section option. You must specify the indicated option. Add the option to the configuration file.
SXF0235E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.
SXF0236E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.
SXF0237E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.
SXF0238E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.
SXF0239E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0241E	FileType cannot be published. Specify Polling or Trigger.	A publishing method was not specified for this file type. Update the configuration file to provide a publishing method for this FileType.
SXF0242E	Only %d message fields options are supported	The maximum number of field options has been exceeded. Correct the configuration file.
SXF0243E	Failed to open TIBCO Rendezvous -- %s	Unable to start TIBCO Rendezvous TIBCO Rendezvous may not be active. Check with your TIBCO administrator that TIBCO Rendezvous is available.

Number	Message	Description
SXF0244E	Failed to create RV transport session: %s	Unable to create a network transport. TIBCO Rendezvous may not be active. Check with your TIBCO administrator that TIBCO Rendezvous is available.
SXF0245E	Failed to open TIB/Rendezvous RVCM -- %s	Unable to start TIBCO Rendezvous. TIBCO Rendezvous may not be active. Check with your TIBCO administrator that TIBCO Rendezvous is available.
SXF0246E	Failed to initialize RV Transport session: %s	Unable to create a network transport session. TIBCO Rendezvous may not be active. Check with your TIBCO administrator that TIBCO Rendezvous is available.
SXF0247E	Failed to initialize RVCM Transport session: %s	Unable to create a transport for certified delivery. TIBCO Rendezvous may not be active. Check with your TIBCO administrator that TIBCO Rendezvous is available.
SXF0248E	Configuration File Error near line <%d> missing ',' or ',' or Contents=%s	There is an error in the configuration file. Verify that the configuration file is coded properly.
SXF0249E	Found definitions for both Reliable and Certified mode	The configuration file contains definitions for both modes. The modes are exclusive. Use only one type of mode.
SXF0250E	Found definitions for both Reliable and Certified mode	The configuration file contains definitions for both modes. The modes are exclusive. Use only one type of mode.
SXF0251E	Missing tag <%s> for %s	The configuration file is missing a required parameter. The error message indicates which parameter is missing. Check the configuration file, and correct the discrepancy.
SXF0252E	Missing tag <%s> for %s	The configuration file is missing a required parameter. The error message indicates which parameter is missing. Check the configuration file, and correct the discrepancy.
SXF0253E	Unable to allocate memory for MaxECMSubs of length=%d	Unable to allocate memory. Report error to your TIBCO administrator.

Number	Message	Description
SXF0256E	Invalid file type: no prefix or extension	The File Type definition requires a prefix= entry to be supplied.
SXF0260E	SFT and Record Mode ECM Transfers require that you specify an endPublishSubject	SFT and Record Mode ECM operation requires that a endPublishSubject= parameter be supplied in the FileType definition.
SXF0261E	Do not use both <position> and <fieldStart> tags	You used both a position and a fieldStart tag in a constraint parameter. Use only one of the tags to specify a field location.
SXF0262E	Do not use both <position> and <fieldStart> tags	You used both a position and a fieldStart tag in a constraint parameter. Use only one of the tags to specify a field location.
SXF0263E	Expecting <%s> or <%s> tag	The configuration file is missing a required position parameter. The error message indicates which parameter is missing. Check the configuration file and correct the discrepancy.
SXF0264E	Expecting <%s> tag	The configuration file is missing a required length parameter. The error message indicates which parameter is missing. Check the configuration file and correct the discrepancy.
SXF0265E	Expecting <%s> tag	The configuration file is missing a required location parameter. The error message indicates which parameter is missing. Check the configuration file and correct the discrepancy.
SXF0266E	Invalid data type <%s> between line <%d> and line <%d>	The "type" value is invalid. Correct the "type" value entered.
SXF0267E	Config parse error between line %d and line %d, invalid precision: <%s>	The precision value is invalid. Ensure that the precision specified in the configuration file is entered correctly. The precision values must be numeric positive values, separated by a comma.

Number	Message	Description
SXF0268E	Config parse error between line %d and line %d, invalid precision: <%s>	The precision value is invalid. Ensure that the precision specified in the configuration file is entered correctly. The precision values must be numeric positive values, separated by a comma.
SXF0269E	Config parse error between line %d and line, bad message token: '%s'	In the configuration file for FileSubscriber, a token is invalid in the FILE_LINE element of the FileType section. The invalid token is displayed. Correct or delete the invalid token.
SXF0270E	Binary fields require a PRECISION definition tag	The configuration file does not have a "precision" tag for this field. Edit the configuration and supply a precision value for all Packed and Zoned fields.
SXF0271E	Expecting <%s> or <%s> tag	The FileAdapter was expecting parameter or tag identified by %s in the configuration, and did not find it. Add that tag to the definition.
SXF0272E	Expecting <%s> tag	The FileAdapter was expecting parameter or tag identified by %s in the configuration, and did not find it. Add that tag to the definition.
SXF0273E	Expecting <%s> tag	The FileAdapter was expecting parameter or tag identified by %s in the configuration, and did not find it. Add that tag to the definition.
SXF0274E	Expecting <%s> tag	The FileAdapter was expecting parameter or tag identified by %s in the configuration, and did not find it. Add that tag to the definition.
SXF0275E	Packed/Zoned/Binary/Float fields require a PRECISION definition tag	The configuration file does not have a "precision" tag for this field. Edit the configuration and supply a precision value for all numeric fields.
SXF0276E	Packed/Binary/Zoned/Float fields require Fixed field width processing	A FileType has a field type defined that requires a "binary" file specification. Specify isBinary="true" for this file type.
SXF0277E	Packed/Binary/Zoned/Float fields can only be defined with BINARY files.	A FileType has a field type defined that requires a "binary" file specification. Specify isBinary="true" for this file type.

Number	Message	Description
SXF0278E	Unable to allocate memory for Msg Container of length=%d	Unable to allocate memory. Report error to your TIBCO administrator
SXF0279E	Config parse error between line %d and line %d, failed to parse value for %s	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF0280E	Config parse error between line %d and line %d, missing container name	A message container does not have a name. Check the configuration file, and make sure that all message containers have a name=" " entry. Values are case sensitive.
SXF0281E	Config parse error between line %d and line %d, bad message token: '%s'	In the configuration file for FileSubscriber, a token is invalid in the FILE_LINE element of the FileType. The invalid token is displayed. Correct or delete the invalid token.
SXF0282E	Option Attributes: Memory re-allocation failed	Unable to allocate memory. Report error to your TIBCO administrator.
SXF0283E	RV Session Attributes: Memory allocation failed, length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF0284E	RV Session Attributes: Config parse error near line %d, failed to parse value	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF0285E	RVCM Session Attributes: Memory allocation failed, length=%d	Unable to allocate memory. Report error to your TIBCO administrator.

Number	Message	Description
SXF0286E	RVCM Session Attributes: Config parse error near line %d, failed to parse value	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF0287E	Config parse error near line %d, invalid option name: '%s'	One of the values in the Options section may be misspelled, or may not have the proper case. The invalid option is displayed. Check the spelling of the values in the Options section of the configuration file. Values must be in upper case
SXF0288E	Config parse error near line %d, failed to parse Options line	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF0289E	FileType Attributes: Memory re-allocation failed	Unable to allocate memory. Report error to your TIBCO administrator
SXF0290E	FileType Attributes: Memory allocation failed, length=%d	Unable to allocate memory. Report error to your TIBCO administrator
SXF0291E	FileType Attributes: Config parse error between line %d and line %d, failed to parse value	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF0292E	MsgFields Attributes: Memory allocation failed, length=%d	Unable to allocate memory. Report error to your TIBCO administrator

Number	Message	Description
SXF0293E	FileType Attributes: Config parse error between line %d and line %d, failed to parse value	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF0294E	Config parse error between line %d and line %d, invalid option name: '%s'	One of the values in the Options section may be misspelled, or may not have the proper case. The invalid option is displayed. Check the spelling of the values in the Options section of the configuration file. Values must be in upper case.
SXF0295E	Config parse error between line %d and line %d, failed to parse File Type line	In the configuration file, the Options section must have matching opening and closing '}'. Check the configuration file and make sure that the Options section has matching opening and closing '}'.
SXF0296E	Config parse error near line %d, invalid Trace option name: '%s'	One of the values in the Options section may be misspelled, or may not have the proper case. The invalid option is displayed. Check the spelling of the values in the Options section of the configuration file. Values must be in upper case.
SXF0297E	Config parse error near line %d, failed to parse Trace line	In the configuration file, the Options section must have matching opening and closing '}'. Check the configuration file and make sure that the Options section has matching opening and closing '}'.
SXF0298E	Failed to pre-register listener %s for subject %s cause=%s	An invalid subject name was given, or there is a problem with TIBCO Rendezvous. Specify a valid subject name. If the problem continues, contact your TIBCO administrator.
SXF0299E	Expecting <%s> tag	The configuration file is missing a required parameter. The error message indicates which parameter is missing. Check the configuration file and correct the discrepancy.

Number	Message	Description
SXF0300E	Expecting <%s> tag	The configuration file is missing a required parameter. The error message indicates which parameter is missing. Check the configuration file and correct the discrepancy.
SXF0301E	PreRegister Attributes: Memory re-allocation failed	Unable to allocate memory. Report error to your TIBCO administrator.
SXF0302E	Only %d Subscribers can be pre-registered	You have specified too many pre-registered subscribers. Reduce the number of pre-registered listeners. The maximum is 126.
SXF0303E	Only %d Subscribers can be pre-registered	You have specified too many pre-registered subscribers. Reduce the number of pre-registered listeners. The maximum is 126.
SXF0304E	Subscriber Attributes: Memory allocation failed, length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF0305E	Subscriber Attributes: Config parse error near line %d, failed to parse value	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF0306E	Config parse error near line %d, invalid Subscriber option name: '%s'	One of the values in the Options section may be misspelled, or may not have the proper case. The invalid option is displayed. Check the spelling of the values in the Options section of the configuration file. Values must be in upper case.
SXF0307E	Config parse error near line %d, failed to parse options line	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF0308E	Only %d file types are supported	A maximum of 1024 file types is supported, for each File Adapter.

Number	Message	Description
SXF0309E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0310E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0311E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0314E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0315E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0316E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0317E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0318E	Process Library not found: %s	The library indicated was not found. Verify that the data set exists and is available to FilePublisher.
SXF0319E	Output Library not found: %s	The library indicated was not found. Verify that the data set exists and is available to FilePublisher.
SXF0320E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0321E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.

Number	Message	Description
SXF0322E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF0323E	Process Directory not found: %s	The specified directory was not found. Make sure that the directory specified exists and is available.
SXF0324E	Output Directory not found: %s	The specified directory was not found. Make sure that the directory specified exists and is available.
SXF0325E	Missing %s definition	No Adapter name has been specified for this configuration. Correct the configuration file.
SXF0326E	Failed to open config file: %s	Unable to open the configuration file or the configuration file was not found. Make sure that the configuration file exists, and that the Adapter can find it.
SXF0327E	Memory allocation for Max FileType failed, length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF0330W	endPublishSubject is not used by Block Mode ECM. Parameter ignored.	endPublishSubject is not a valid parameter to use when defining Block Mode ECM file types.
SXF0331W	Invalid value <%s> for DELETE_PUB_PRG_FILES. Parameter ignored.	The value specified for the DELETE_PUB_PRG_FILES. Is not correct. Go re-check the correct options in the manual.
SXF0333E	File <%.44s> not found. Requested by user=% on Subject/Destination %s. File publish request Rejected.	The File Adapter cannot find a valid entry for this file. It is either mis-spelled or has been deleted.
SXF0336W	Both Code Pages must be configured. Single code page config is ignored.	Normally, both HOSTCODEPAGE and NETWORKCODEPAGE should be specified when overriding the default code pages used by the File Adapter.

Number	Message	Description
SXF0337W	Code Page Setup failed: reason=<%s>. Will use defaults instead.	Unable set up the requested code pages. Verify that the code page settings are valid for your system and that TIBCO Rendezvous is active. Check with your TIBCO administrator.
SXF0339E	Location value (%d) exceeds max allowed (%d), near line %d	The location= attribute tag exceeded the maximum field offset allowed in a record. Check the configuration file, and correct the discrepancy.
SXF0340E	Total number of message item/container definitions (%d) exceeds max number of items allowed (%d), near line %d	The maximum number of message definitions for a given FileType has been exceeded. Correct the configuration file.
SXF0342E	Missing Block Mode ECM Subscriber defs. FileType definition ignored	Block Mode ECM file type definitions require a EcmSubscriber definition. The Block mode definition is incorrect, because it lacks any EcmSubscriber definitions.
SXF0343E	Missing Record Mode ECM confirmationSubject/endPublish Subject or TotalCount defs.	Record Mode ECM file type definitions require a confirmationSubject and a Total count. Supply those parameters.
SXF0344E	Missing useExplicitConfirmation defs when using %s. FileType definition ignored	A config mismatch has been detected. When using the parameter identified by %s, a useExplicitConfirmation must also be used.
SXF0345E	Config error near line %d ic=%d:\n contents=%s	A configuration error was detected in the INI file near the line specified. The %s contains the reason for the error.
SXF0346E	Config error near line %d. Reading till next section is encountered	A configuration error was detected in the INI file near the line specified. The File Adapter will skip to the next section, because the current definition has too many severe config errors.
SXF0363E	Failed to create connection to the server. Reason :%s, exiting...	RV was unable to create the connection needed to communicate. The %s identifies the error it encountered.

Number	Message	Description
SXF0374E	Invalid %s setting=%s. Must be RecordMode, RecordModeECM, BlockModeSFT, or BlockModeECM	Valid values for the transferType parameter are RecordMode, RecordModeECM, BlockModeSFT, or BlockModeECM.
SXF0381E	EMS protocol is not allowed with ECM.	The EMS protocol is not allowed with ECM.
SXF0383E	Keyword %s is not allowed with %s protocol.	The specified keyword is not allowed with the specified protocol.
SXF0385E	EMS acknowledgement %s not allowed; changed to %s.	The specified EMS acknowledgement is not allowed. It has been changed to the value reported.
SXF0387E	EMS acknowledgement %s is invalid.	The specified EMS acknowledgement is not valid.
SXF0389E	TOPIC required for EMS durable subscriber %s.	The EMS durable subscriber requires a destination type of TOPIC.
SXF0390W	startNewMessage set to default value: <true>	For EMS, a startNewMessage value of <false> is not allowed. It has been set to the only allowed value <true>.

SXF1000 – SXF1999

Number	Message	Description
SXF1011E	Failed to unsubscribe for trigger Subject/Destination <%s>: reason=%s	The Adapter was being shut down, and it encountered an error when trying to disconnect. The %s identifies the error it encountered.
SXF1013E	Failed to unsubscribe for trigger Subject/Destination <%s>: reason=%s	The Adapter was being shut down, and it encountered an error when trying to disconnect. The %s identifies the error it encountered.

Number	Message	Description
SXF1045E	Failed to append %s to Job Queue file - storing in memory only	Unable to append an incoming Trigger request to the on-disk job queue file. Check the configuration file to ensure a valid job queue filename was set up.
SXF1046E	Subject/Destination name %s does not match file type %d	A trigger was received for a file type that does not support the requested subject destination.
SXF1047E	Failed to update Job Queue file - storing in memory only	Unable to update a completed request in the on-disk Job Queue file. Check the configuration file to ensure a valid Job Queue filename was setup.
SXF1048E	Failed to process trigger for Subscriber %s, using fieldnames=%s and %s error message: %s	A trigger message was received that did not have the minimum required parameters (e.g. a field name of "DATA" or "fileName", or a user-defined "Trigger" file name). FileAdapter is unable to process the message. Fix the application that is sending the trigger message to include the required field.
SXF1050E	MAXRECORDS field %s on trigger message to %s is an invalid number. Rejected	FilePublisher received a trigger message, but was unable to process it because the specified parameter was invalid. Check that the input value for the parameter is correct, check that TIBCO Rendezvous is available, and re-submit the trigger message.
SXF1053E	Security Check failed. UserId=<%s> is not allowed READ access to File=<%s> from Subject/Destination=<%s>. rc=%d. Trigger request rejected, File not published.	Security checking was requested, and the security facility indicated the incoming Trigger request from the specified user, was not allowed access to publish that file. Contact your security people, to determine why the user is not authorized to publish that file.
SXF1055E	Trigger request for file %s rejected because no UserId received on subject %s.	FilePublisher received a trigger message, but was unable to process it because Security checking is turned on, and no user id was included on the trigger message. FileAdapter is unable to process the message. Fix the application that is sending the trigger message to include the required field.

Number	Message	Description
SXF1056E	Security check failed for userId=<%s> File=<%s> from Subject/Destination=<%s>. Return code=<%d>.	Security checking was requested, and the security facility indicated the incoming Trigger request from the specified user, was not allowed access to publish that file. Contact your security people, to determine why the user is not authorized to publish that file.
SXF1508W	RV DAEMON disconnected, exiting app...	The RV Daemon being used by the FileAdapter disconnected, either due to a network crash or a crash of the daemon. The FileAdapter is unable to publish any more messages until the daemon (or its connection) is restored.
SXF1059E	<%s> CONFIG out of sync. invalid job queue index: index=%d max=%d addr=%x items=%d fname=<%s> subjectName=<%s> buff=<%s>	Recovered job queue has a bad entry.
SXF1509W	RV DAEMON disconnected, exiting app...	The RV Daemon being used by the FileAdapter disconnected, either due to a network crash or a crash of the daemon. The FileAdapter is unable to publish any more messages until the daemon (or its connection) is restored.
SXF1511W	%s exited with errors	The File Adapter terminated due to the error specified by the %s reason code.
SXF1521E	Error %d on Heartbeat timer create: %s	The File Adapter was unable to create a Heartbeat timer. The %d and %s indicate what the error was that caused the problem.
SXF1524E	Error %d on Started Task console check timer create: %s	The File Adapter was unable to create a Console timer. The %d and %s indicate what the error was that caused the problem.
SXF1526E	Failed to create RV Heartbeat message: %s	Unable to create a heartbeat message. Make sure that TIBCO Rendezvous is available.
SXF1527E	Failed to add [%s] to Heartbeat message: %s	Unable to add the specified string to a heartbeat message. Make sure that TIBCO Rendezvous is available.

Number	Message	Description
SXF1528E	Failed to add [%s] to Heartbeat message: %s	Unable to add the specified string to a heartbeat message. Make sure that TIBCO Rendezvous is available.
SXF1529E	Failed to add file prefix for: %d:%s	Unable to add the file prefix field to a heartbeat message. Make sure that TIBCO Rendezvous is available.
SXF1530E	Failed to add file extension for %d:%s	Unable to add the file extension to a heartbeat message. Make sure that TIBCO Rendezvous is available.
SXF1531E	Failed to add status for: %d:%s	Unable to add the status string to the heartbeat timer message. Check with the TIBCO administrator.
SXF1532E	Failed to add numPublished	Unable to add the numPublished string to the heartbeat timer message. Check with the TIBCO administrator.
SXF1533E	Failed to add progessLineNo	Unable to add the progessLineNo string to the heartbeat timer message. Check with the TIBCO administrator.
SXF1534E	Failed to add progessLineNo	Unable to add the progessLineNo string to the heartbeat timer message. Check with the TIBCO administrator.
SXF1535E	Failed to set send Heartbeat target: cause=%s Subject/Destination=%s	An error was returned when the Adapter attempted to set the subject name in a heartbeat message to be published. Ensure that a valid subject name is specified in the configuration file.
SXF1536E	Error in sending EMS/RV Heartbeat message: cause=%s Subject=%s	Unable to publish a heartbeat message. Make sure that TIBCO Rendezvous is available.
SXF1537E	Failed to destroy message.	Unable to destroy a published heartbeat message. Make sure that TIBCO Rendezvous is available.

Number	Message	Description
SXF1538E	Configuration file from -config parameter and QINLINE file not found. Will use default config filename <%s> instead.	No configuration file was passed in as a parameter and no inline data file was found. This is probably an error by the operator. Double-check that you really want to use the default configuration file.
SXF1539E	Failed to get hostname from system. Error code returned by 'gethostname' is = %d	Unable to extract the host name being used by TCPIP. Probably indicates that TCP/IP is not up or not functioning.
SXF1540E	Could not find a usable TCP service.	A missing TCP Steplib entry is causing problems.
SXF1541E	Failed to get hostname from system. Error code returned by 'gethostname' is = %d	The TCP Comm Server is unable to return the hostname of the current system. Contact the system programmer to try to determine what the problem.
SXF1542E	Error in sending EMS/RV Heartbeat message: cause=%s subject=%s	Unable to publish a heartbeat message. Make sure that TIBCO EMS/Rendezvous is available.
SXF1543E	FilePoller List Memory allocation failed	Unable to allocate memory. Report error to your TIBCO administrator.
SXF1544E	Failed to subscribe to Subject/Destination <%s>: %s	Unable to create a listener. TIBCO Rendezvous may be unavailable or may not be active. Check with your TIBCO administrator.
SXF1545E	Failed to subscribe to Subject/Destination <%s>: %s	Unable to create a listener. TIBCO Rendezvous may be unavailable or may not be active. Check with your TIBCO administrator.
SXF1546E	FilePoller Memory allocation failed for length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF1547E	No file type objects were created from configuration	Unable to create a file type object. At least one file type entry should be completely defined in the configuration file.
SXF1548E	JobQueue element Memory allocation failed for length=%d	An out of memory error was reported.

Number	Message	Description
SXF1549E	JobQueue element Memory allocation failed for length=%d	An out of memory error was reported.
SXF1550E	TriggerQue Memory allocation failed for length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF1551E	TriggerQue Memory allocation failed for length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF1552E	File Poller Timer removal failed: %s	The File Adapter encountered an error while terminating. The %s details the reason for the problem.
SXF1553E	Max_Dir FileList Memory allocation error for length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF1554E	Max_Dir FileList Memory allocation error for length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF1555E	File polling for %s data set is not supported!	The data set type is invalid for polling. The only data set that is supported for polling is SEQ. Use a different method to publish the file.
SXF1556E	Timer creation failed: %s	Unable to create a timer. This is a system-level error. Check that TIBCO Rendezvous is available and functioning properly. Contact the TIBCO administrator.
SXF1557E	FilePoller's Memory allocation of FileSender struct failed for length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF1558E	NULL filePoller in timer callback	The Adapter attempted to create a callback function, but the timer callback function is NULL. This is a system-level message. Contact your TIBCO administrator.
SXF1559E	Failed to subscribe to Subject/Destination <%s>: cause=%s	The File Adapter was unable to successfully subscribe to a given Subject. The %s details the reason for the problem.

Number	Message	Description
SXF1560E	Failed to subscribe to Subject/Destination <%s>: cause=%s	The File Adapter was unable to successfully subscribe to a given Subject. The %s details the reason for the problem.
SXF1564E	Transaction Delay Timer removal failed: %s	The Adapter was unable to remove the transaction delay timer.
SXF1565E	Admin Timer removal failed: %s"	The Adapter was unable to remove the ECM Admin message timer

SXF2000 – SXF2999

Number	Message	Description
SXF2004E	<%s> Unable to locate Subscriber=%s	A confirmation reply was received, but no such subscriber was found in the current configuration. It indicates some sort of configuration mis-match between the Publisher and the Subscriber.
SXF2005E	<%s> No subscriber name found on Confirm msg.	A confirmation reply was received, but no such subscriber was found in the current configuration. It indicates some sort of configuration mis-match between the Publisher and the Subscriber.
SXF2010E	Close publisher file=<%s> failed.	A close on a file being published failed. Save the %s reason code and discuss the issue with your systems programmer.
SXF2023W	ExecuteAfterProcess ignored because the file <%s> has been removed	An ExecuteAfterProcess was requested in the configuration, but the target file was removed/deleted, and not available for use.
SXF2050W	<%s> Re-publishing EOF End Message to Subject/Destination=<%s> Block=%d idx=%d retry=%d #confBlks=%d isEOFblk=%d GUID=%s	The File Publisher did not receive a Block Mode or Record Mode ECM acknowledgement for the last block it sent. If this problem persists it may indicate a network problem. The Publisher will retry to send the block again.

Number	Message	Description
SXF2080E	Close publisher file: <%s> failed.	A close on a file being published failed. Save the %s reason code and discuss the issue with your systems programmer.
SXF2096W	<%s> Re-publishing to Subject/Destination=<%s> %s=%d idx=%d retry=%d #confBlks=%d GUID=%s Eofsm=%d empty=%d	The File Publisher did not receive a Block Mode or Record Mode ECM acknowledgement for the last block it sent. If this problem persists it may indicate a network problem. The Publisher will retry to send the block again.
SXF2103W	Move operation could not remove file <%s>	Source file is retained while moving a file. Source file was in use by another process when the Adapter attempted to move the file. Remove the source file if it is not needed.
SXF2104W	Libraries are identical, file was not moved.	One of the input, process, or output libraries was not unique. Specify input, process and output libraries that are unique.
SXF2105W	Opening File <%s> in Binary Mode %s lineLength=%d	A file is being opened in Binary Mode with a linelength specified that may not match the file's actual length.
SXF2106W	<%s> Move operation could not remove source file. cause=%d: %s	A remove operation on a file that was published failed. Save the %s reason code and discuss the issue with your systems programmer.
SXF2107W	Libraries are identical, file is not moved.	The indicates a configuration mis-match between the input or process files, and the output target files.
SXF2108W	Failed to remove file: <%s> because there were errors during processing.	Unable to remove the file indicated. The file still exists on the system. Verify that FilePublisher has the proper authority to remove files.
SXF2109W	<%s> Failed to remove 'AfterProcess' file=<%s>. Cause=%d %s	Unable to remove the file indicated. The file still exists on the system. Verify that FilePublisher has the proper authority to remove files.

Number	Message	Description
SXF2110W	<%s> Failed to remove file=<%s>. Cause=%d %s	Unable to remove the file indicated. The file still exists on the system. Verify that FilePublisher has the proper authority to remove files.
SXF2111W	Failed to extract nested msg. field <%s> to form Subject/Destination name.	Unable to construct the subject from the field data. Check the configuration file for this message.
SXF2112W	Failed to extract field <%s> to form Subject/Destination name.	Unable to construct the subject from the field data. Check the configuration file for this message.
SXF2113W	<%s> File pointer is NULL in publishBlock invoked from timer callback	The file pointer for a timer function is NULL. Report this to the TIBCO administrator. This portends a system error.
SXF2114W	File pointer is NULL in pubRecord callback. No file was open. pubRecord callback invoked. No file is open currently.	The file pointer for a timer function is NULL. Report this to the TIBCO administrator.
SXF2115W	Failed to remove file: <%s> because there were errors during processing.	Unable to remove the file indicated. The file still exists on the system. Verify that FilePublisher has the proper authority to remove files.
SXF2116W	Failed to remove file: <%s> cause=%s.	Unable to remove the file indicated. The file still exists on the system. Verify that FilePublisher has the proper authority to remove files.
SXF2117W	Failed to remove file: <%s> cause=%s.	The file still exists on the system. Verify that FilePublisher has the proper authority to remove files.
SXF2118W	<%s>: line %d has no fields	Configuration error. No fields were found in the file specified. Correct the configuration file.
SXF2119W	The specified constraint could not be found in the input file <%s>	Configuration error. The specified constraint could not be found in the input file. Correct the configuration file.

Number	Message	Description
SXF2120E	Move failed: could not open file <%s>	The move operation failed. Check that the target directory is available and that there are no authority problems.
SXF2121E	Move failed: could not open output file <%s>	The move operation failed. Check that the target directory is available and that there are no authority problems.
SXF2122E	<%s> Move failed: could not open source file using mode=%s. cause=%d: %s	The move operation failed. Check that the target directory is available and that there are no authority problems.
SXF2123E	<%s> Move failed: could not open target file using mode=%s. cause=%d: %s	The move operation failed. Check that the target directory is available and that there are no authority problems.
SXF2124E	<%s> Move failed: Unable to close source file cause=%d: %s	The file specified could not be moved. Be sure that FilePublisher has the proper authority.
SXF2125E	<%s> Move failed: Write error on output file. cause=%d: %s	Unable to move a file. Ensure that FilePublisher has the authority to move the specified file.
SXF2126E	<%s>Move failed: Data loss while flushing the output file	Unable to complete moving a file. Ensure that FilePublisher has the authority to move the specified file.
SXF2127E	<%s> Move failed: Error renaming input file	The File Adapter was unable to successfully rename a CWK file to the target filename used by the FileType. The second %s details the reason for the problem.
SXF2128E	<%s> Move failed: Error moving Binary input file.	The File Adapter was unable to successfully rename a CWK file to the target filename used by the FileType. A second related error message will detail why the move failed.
SXF2129E	<%s> Move failed: Error moving Text input file.	The File Adapter was unable to successfully rename a CWK file to the target filename used by the FileType. A second related error message will detail why the move failed.

Number	Message	Description
SXF2130E	<%s> Move failed: Error renaming input file cause=%d: %s	The File Adapter was unable to successfully rename a CWK file to the target filename used by the FileType. The second %s details the reason for the problem.
SXF2131E	Failed to extract message <%s> for adding container: cause=%s	Possible mismatch between the received message and the configuration file.
SXF2132E	Failed to Add message <%s> to container: cause=%s	Possible mismatch between the received message and the configuration file.
SXF2133E	Failed to Update message <%s> to container: cause=%s	Unable to update one or more fields in a container message. Check the configuration file. The message may have exceeded the maximum buffer length, or TIBCO Rendezvous may be unavailable.
SXF2134E	Failed to Add message <%s> to container: cause=%s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2135E	Failed to Add String to Data message: cause=%s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2136E	Failed to get message description tag: %s	Probably a configuration mismatch. Check the configuration file and ensure that the configuration matches the received message.
SXF2137E	Failed to Add Opaque/Binary to Data message: cause=%s	An attempt to add a field to an RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory for the REGION. If increasing the REGION size does not solve the problem, then contact TIBCO technical support.
SXF2138E	Packed decimal data is invalid for field <%s>. Zeros sent	The data in a PACKED field is not valid. Ensure that the file data being published is of the right format.
SXF2139E	Packed decimal data is invalid for field <%s>. Zeros sent	The data in a PACKED field is not valid. Ensure that the file data being published is of the right format.

Number	Message	Description
SXF2140E	Failed to Add Packed to Data message: cause=%s	Unable to add the specified field to a data message. Check with the TIBCO administrator.
SXF2141E	Zoned field <%s> contains invalid character <%c>. Zeros sent	A ZONED field contains invalid data. Check the file data to ensure that it is of the proper type.
SXF2142E	Failed to Add Zoned to Data message: cause=%s	Unable to add the specified field to a data message. Check with the TIBCO administrator.
SXF2143E	Failed to destroy timer: cause=%s	Unable to destroy a timer after publishing a message. Check with the TIBCO administrator.
SXF2145E	Failed to unsubscribe to confirmation Subject/Destination: cause=%s	During shutdown, the FileSubscriber was unable to terminate the subscription to a Subject. The cause %s specifies what the problem was
SXF2147E	<%s> Command [%s] returned error rc=%d	The command identified did not complete successfully. Check the command specified for executeBeforeProcess or executeAfterProcess and ensure that the commands are valid.
SXF2150E	<%s> Command [%s] returned error rc=%d	The command identified did not complete successfully. Check the command specified for executeBeforeProcess or executeAfterProcess and make sure that the commands are valid.
SXF2152E	<%s> Remote Confirmation message indicates Subscriber I/O error. Terminating Block transfer to Sub=%s. Remaining Active Subs=%d.	A Block Mode ECM subscriber indicated that it received an out-space error, or some other kind of I/O error when writing to the work (CWK) file. The File Adapter will drop that subscriber and continue to publish to any other active subscriber.

Number	Message	Description
SXF2153E	<%s> Remote Confirmation message indicates Subscriber I/O error. Terminating Block transfer to Sub=%s.	A Block Mode ECM subscriber indicated that it received an out-space error, or some other kind of I/O error when writing to the work (CWK) file. There are not other active subscribers for that File, so the transfer is being terminated
SXF2154E	<%s> Failed to retrieve GUID from Confirmation message	An attempt to retreive a field to an RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory for the REGION. If increasing the REGION size does not solve the problem, then contact TIBCO technical support.
SXF2155E	<%s> Failed to retrieve Block Number from Confirmation message	An attempt to add a field to an RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory for the REGION. If increasing the REGION size does not solve the problem, then contact TIBCO technical support.
SXF2156E	<%s> processing file <%s> set Subscriber %s offline. Its Max FlexECM retry limit was reached.	A Block Mode ECM subscriber exceed its maximum error count thresh-hold, and has been dropped from the list of active subscribers.
SXF2157E	<%s> set Offline while sending file <%s>. All active ECM Subscribers failed, and Max FlexECM retry limit reached.	All the subscribers to a Block Mode ECM transfer have timed out. This normally indicates some sort of network problem.
SXF2160E	<%s> Failed to close .PRG file=%s, aborting recovery	Unable to close the progress file. Check the process library. The file may be corrupted.
SXF2161E	<%s> Failed to remove .PRG file=%s, aborting recovery	Unable to remove the progress file. Check the process library. The file may be corrupted or not properly cataloged.
SXF2162E	<%s> Failed to read data in progress file=%s. Wrong number of fields (%d), aborting recovery	Unable to read the progress file. Check the process library. The file may be corrupted.
SXF2163E	<%s> Failed to close .PRG file=%s, aborting recovery	Unable to close the progress file. Check the process library. The file may be corrupted.

Number	Message	Description
SXF2164E	<%s> Failed to remove .PRG file=%s, aborting recovery	Unable to remove the progress file. Check the process library. The file may be corrupted or not properly cataloged.
SXF2165E	<%s> Failed to read data in progress file=%s. Read of file failed, aborting recovery	Unable to read the progress file. Check the process library. The file may be corrupted.
SXF2166E	<%s> Failed to close .PRG file=%s, aborting recovery	Unable to close the progress file. Check the process library. The file may be corrupted.
SXF2167E	<%s> Failed to remove .PRG file=%s, aborting recovery	Unable to remove the progress file. Check the process library. The file may be corrupted or not properly cataloged.
SXF2168E	<%s> Failed to extract nested RV message for NT_ECM TEXT for ECM publish block. Reason: %s	FilePublisher was unable to extract the nested TEXT data field from a ECM block data message. Check with the TIBCO administrator.
SXF2169E	<%s> Failed to create RV message of length=%d for ECM publish block. Reason: %s	Unable to create a TIBCO Rendezvous message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2170E	<%s> Failed to create RV message of length=%d: Reason: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2171E	<%s> Failed to add %s to message: %s	Unable to create a message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2172E	<%s> Failed to create nested RV message of length=%d: Reason: %s	Unable to create a nested message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2173E	<%s> Failed to update nested RV message of length=%d: Reason %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2174E	<%s> Failed to add %s to RV message: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.

Number	Message	Description
SXF2175E	<%s> Failed to add %s to RV message: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2176E	<%s> Failed to add %s to RV message: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2177E	<%s> Failed to add %s to RV message: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2178E	<%s> Failed to add %s to RV message: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2179E	<%s> Failed to add %s to RV message: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2180E	<%s> Failed to add %s to RV message: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2181E	<%s> Failed to create EndMsg RV message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is up and running.
SXF2182E	<%s> Failed to add %s to RV End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2183E	<%s> Failed to add Filename to RV End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2184E	<%s> Failed to add NumberOfMessage to RV End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2185E	<%s> Failed to add %s to RV End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.

Number	Message	Description
SXF2186E	<%s> Failed to add %s to RV End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2187E	<%s> Failed to add %s to End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2188E	<%s> Failed to add %s to End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2189E	<%s> Failed to add %s to End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2190E	<%s> Failed to add %s to End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2191E	<%s> Failed to add %s to End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2192E	<%s> Failed to add %s to End message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2193E	Failed to create Error RV message of length=%d: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2194E	Failed to add [%s] to Error message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2195E	Failed to set Error send subject/Destination: %s cause=%s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2196E	Failed to send error message: cause=<%s> Subject/Destination=%s	File Publisher was unable to send a message to the network. The %s cause details why. It may indicate a network or RV daemon problem.

Number	Message	Description
SXF2197E	Failed to destroy Error RV message: reason=%s	File Publisher was unable to send a message to the network. The %s cause details why.
SXF2198E	Cannot open null file name. rmode=%d	File name in configuration file was NULL (""). Check the configuration file, and supply a file name where it's missing.
SXF2201E	<%s> File open failed for file <%s> using mode '%s'. errno=%d R15=%d Fdbk=%d op=%d ab_sys=0x%X ab_rc=0x%X\n errno text=%s	Unable to open the file identified in the message. Correct the configuration file entry for this file, and be certain that the file is available for use.
SXF2206E	<%s> Can not open file <%s> because OPEN flag still set on from previous run or another App is trying to share the file.	Unable to open the file identified in the message. Correct the configuration file entry for this file, and ensure that the file is available for use.
SXF2209E	<%s> Failed to open file '%s', errno=%d.	Unable to open the file identified in the message. Correct the configuration file entry for this file, and ensure that the file is available for use.
SXF2210E	<%s> Transfer aborted for file: <%s>. LRECL length %d > blockTransferSize %d	Unable to process the file identified in the message because the LRECL of the file was longer than the ECM block size specified for the transfer. Correct the configuration file entry for this file to make blockTransferSize equal to or larger than the file's LRECL.
SXF2211E	Failed to open .PRG file '%s' errno=%d errtext=%s	Unable to open the progress .PRG file identified in the message. Check to see that the .PRG file is available for use.
SXF2212E	<%s> TransactionDelay Timer creation failed: %s	Unable to create a timer. This is a system level error. Check that TIBCO Rendezvous is available and functioning properly. Contact the TIBCO administrator.
SXF2213E	<%s> ConfBlock - Failed to create message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is up and running.

Number	Message	Description
SXF2214E	Failed to set RV send destination/ subject: %s	An error was returned when the Adapter attempted to publish a message. Ensure that a valid destination/subject name is specified in the configuration file.
SXF2215E	Failed to set RV reply destination/ subject: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory for the REGION. If increasing the REGION size does not solve the problem, then contact TIBCO technical support.
SXF2216E	Error in sending RVCM message: cause=%s Subject/Destination=%s	An error occurred when sending an RVCM message. The Adapter may not be active. Contact your TIBCO administrator and report this error.
SXF2217E	<%s> Error in sending RV message: cause=%s Subject=%s.	An error occurred when sending a RV message. The Adapter may not be active. Contact your TIBCO administrator and report this error.
SXF2218E	<%s> Failed to create RV message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2219E	<%s> Failed to initialize RV message of length=%d: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2220E	<%s> Failed to add %s to message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2221E	<%s> Failed to add %s to message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2222E	<%s> Failed to destroy rvMsg: %s OR <%s> Failed to destroy emsMsg: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.

Number	Message	Description
SXF2223E	<%s> Failed to remove timer: %s	Unable to terminate an RV timer. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2224E	<%s> Failed to create message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2225E	<%s> Failed to create message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2226E	<%s> Cannot append to message, failed to find start	Unable to append a field onto an existing message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2227E	<%s> Failed to create message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2228E	<%s> Failed to append container: %s cause=%s	Unable to append a field to an existing message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2229E	<%s> Failed to create message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2231E	Failed to destroy message: %s	Unable to free memory. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.
SXF2232E	<%s> Failed to create Start message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2233E	<%s> Failed to append FileName parm to Start message: %	Unable to add the filename field to the Start File Publish message. Check the configuration file to ensure that a valid file name was specified. Make sure that TIBCO Rendezvous is available.

Number	Message	Description
SXF2234E	<%s> Failed to create Tracking message of length=%d: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2235E	<%s> Failed to create Trigger Reply message of length=%d: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2236E	<%s> Failed to append fields to Trigger Reply message: %s	Unable to add one or more fields to a Trigger Reply message. Make sure that TIBCO Rendezvous is available.
SXF2237E	<%s> Failed to create NT_ECM Listener: %s	Unable to create a RV Listener. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2238E	<%s> Failed to add String Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2239E	<%s> Failed to add String Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2240E	<%s> Failed to add Bool Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2241E	<%s> Failed to add Time Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2242E	<%s> Failed to add I16 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.

Number	Message	Description
SXF2243E	<%s> Failed to add U16 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2244E	<%s> Failed to add I32 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2245E	<%s> Failed to add U32 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2246E	<%s> Failed to add F32 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2247E	<%s> Failed to add F64 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and ensure that the configuration matches the message to be sent.
SXF2248E	<%s> Field %s: Invalid data type %s (%d)	An invalid message description was encountered. Check that the configuration file matches the message received, and that TIBCO Rendezvous is available.
SXF2249E	<%s> Invalid length defined for string field: %s	An invalid message description was encountered. Check that the configuration file matches the message received, and that TIBCO Rendezvous is available.
SXF2250E	<%s> Failed to add String Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2251E	<%s> Failed to add String Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.

Number	Message	Description
SXF2252E	<%s> Failed to add String Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2253E	<%s> Failed to add Bool Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2254E	<%s> Failed to add Time Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2255E	<%s> Failed to add I16 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2256E	<%s> Failed to add U16 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2257E	<%s> Failed to add I32 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2258E	<%s> Failed to add U32 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2259E	<%s> Failed to add F32 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.
SXF2260E	<%s> Failed to add F64 Data Field to message at col %d: %s	This is either a configuration mismatch or an out of memory problem. Check the configuration file, and be certain that the configuration matches the message to be sent.

Number	Message	Description
SXF2261E	<%s> Field %s: Invalid data type %s (%d)	An invalid message description was encountered. Check that the configuration file matches the message received, and that TIBCO Rendezvous is available.
SXF2262E	<%s> Failed to create message of length=%d: %s	Could not create a message. Check that TIBCO Rendezvous is available.
SXF2263E	<%s> Failed to get container <%s>: %s	Probably a configuration mismatch. Check the configuration file, and be certain that the configuration matches the received message.
SXF2264E	<%s> Failed to update child message %s	Unable to update one or more fields in a message to be published. Check the configuration file, the message may have exceeded the maximum buffer length, or TIBCO Rendezvous may be unavailable.
SXF2265E	<%s> Failed to destroy child message %s	Unable to free memory. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2266E	<%s> Move failed: Unable to close target file. cause=%d: %s	The file specified could not be moved. Be sure that FilePublisher has the proper authority.
SXF2267E	<%s> Move failed: Unable to close source file. cause=%d: %s	The file specified could not be moved. Be sure that FilePublisher has the proper authority.
SXF2268E	Move failed: copying source <%s> to target <%s>	Could not move the file specified to the directory indicated. Ensure that the file and directory exists, and that FilePublisher has the proper authority.
SXF2269E	<%s> Block Buffer: Memory allocation failed for length=%d	Unable to allocate memory. Report this error to your TIBCO administrator.
SXF2270E	<%s> ConfEndMsgAck Memory allocation failed for %d entries	Unable to allocate memory. Report this error to your TIBCO administrator.
SXF2271E	<%s> Confirm Listen EventId Memory allocation failed for %d entries	Unable to allocate memory. Report this error to your TIBCO administrator.

Number	Message	Description
SXF2272E	<%s> ConfBlocks Memory allocation failed for length=%d	Unable to allocate memory. Report this error to your TIBCO administrator.
SXF2273E	<%s> ConfBlock.MsgAck Memory allocation failed for length=%d	Unable to allocate memory. Report this error to your TIBCO administrator.
SXF2274E	Failed to send error message: cause=<%s> Subject/Destination=%s	Unable to send a message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2275E	<%s> Block Buffer: Memory allocation failed for length=%d	Unable to allocate memory. Report this error to your TIBCO administrator.
SXF2276E	Error in sending RVCM message: cause=%s Subject/Destination=%s	Unable to send a message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2277E	<%s> Error in sending message: cause=%s Subject/Destination=%s	Unable to send a message. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2278E	NULL fileSender in timer callback	Attempted to create a callback function, but the timer callback function is NULL. This is a system level error message, contact your TIBCO administrator.
SXF2279E	Failed to subscribe to Subject/Destination <%s> cause=%s	Unable to create a Subscriber for that destination. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2280E	<%s> Failed to subscribe to Subject/Destination <%s> cause=%s	Unable to create a Subscriber for that destination. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.
SXF2281E	<%s> no confirmation Subject/Destination specified for Record Mode ECM	Unable to create a Subscriber for that destination. Contact the TIBCO administrator to be certain that TIBCO Rendezvous is functioning.

Number	Message	Description
SXF2282E	Failed to Add Double to Data message: cause=%s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2285E	<%s> Failed to add fileTransferDuration to Status message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2286E	There is an error in loading the configuration for this file type. Cannot publish file <%s>.	An error was discovered for the FileType associated with this subject, during configuration processing, and the file type was marked invalid and offline. Fix the error in the configuration file, and re-submit the publish request.
SXF2294W	<%s> Re-publishing Queued Messages to Subject/Destination=<%s>: CurrBlk=%d CurrRec=%d RecovBlk=%d retry=%d #confBlks=%d Eofsm=%d ifactive=%d NtEcm=%d	The File Publisher did not receive a Block Mode or Record Mode ECM acknowledgement for the last block it sent. If this problem persists it may indicate a network problem. The Publisher will retry to send the block again.
SXF2296E	Zoned field <%s> contains invalid character <%c>. Zeros sent	The data in the file to be published is invalid. Verify that the message definition is correct, and that the file is correctly formatted.
SXF2313W	<%s> Message acknowledged. Re-Publishing successful on Subject/Destination=%s after %d retries	Indicates that blocks that were previously being re-published have now been acknowledged. Whatever network condition was causing the problem has been corrected.
SXF2323E	<%s> Failed to create nested EPM Payload message of length=%d: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2326E	<%s> Failed to read data from input process file. Next Block # %d	An attempt to retrieve a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF2327E	Error adding field to EPM message: cause=%s Subject/Destination=%s	An attempt to add a field to a RVmessage failed. The %s identifies the specific cause. It normally indicates insufficient memory.

Number	Message	Description
SXF2342E	Invalid null pointer for pFileSender. Terminating Publisher	An unexpected condition has been detected. The Publisher will be terminated.
SXF2343E	Invalid null pointer for pFileOptions. Terminating Publisher	An unexpected condition has been detected. The Publisher will be terminated.
SXF2344E	Invalid null pointer for %s. pFileSender=%08X timer_kick=%d	An unexpected condition has been detected.
SXF2346E	<%s> Opening file=<%s> with mode=%s failed.	The file adapter was unable to open the specified file.
SXF5329E	Only one durable subscriber can be defined per FileOption section	Only one durable subscriber can be defined per FileOption section.
SXF5332E	Subscribe <%s> and Generate <%s> Destination names need to be equal for Guaranteed Delivery.	If guaranteed delivery of EMS messages is used and if generateFileDestinationName is specified, then subscribeDestinationName must equal generateFileDestinationName

SXF3000 – SXF3999

Number	Message	Description
SXF3000E	<%s> Unable to get reply Subject/Destination for Subscriber=%s	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF3003E	<%s> Unable to locate Subscriber=%s. Not registered.	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF3004E	<%s> No subscriber name found in msgType=%s msg. Unable to reply	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.

Number	Message	Description
SXF3005W	<%s> Ignoring Activate from Subscriber=%s until current file completed.	A recovery race occurred. When the current file is complete, the recovery sequence will be completed.
SXF3008E	<%s> Received unknown ECM Admin Subscriber request %d	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF3009W	<%s> No reply subject/Destination for Subscriber=%s on msgtype=%d.	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF3012E	<%s> Unable to locate Subscriber=%s. Not registered.	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF3013E	<%s> No subscriber name found on msgType=%d. Not registered.	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF3015E	<%s> Received unknown ECM Admin Subscriber reply %d.	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF3021E	<%s> Admin Retry Timer create failed: %s	An attempt to create a RV timer failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF3022E	<%s> set Subscriber %s offline. It's Max FlexECM retry limit was reached during Activation.	A Block Mode ECM subscriber exceed its maximum error count thresh-hold, and has been dropped from the list of active subscribers.
SXF3023E	<%s> set Offline. Not all ECM Subscribers Active, Max FlexECM retry limit reached, and .PRG file <%s> failed.	A Block Mode ECM subscriber exceed its maximum error count thresh-hold, and has been dropped from the list of active subscribers.
SXF3024E	<%s> set Offline. No ECM Subscribers Active, and Max FlexECM retry limit reached.	All the subscribers to a Block Mode ECM transfer have timed out. This normally indicates some sort of network problem.

Number	Message	Description
SXF3025E	<%s> %d subscribers set Offline, because Max FlexECM retry limit reached.	A group of subscribers to a Block Mode ECM transfer have timed out. This normally indicates some sort of network problem.
SXF3026E	<%s> Failed to create RV Admin message: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF3027E	<%s> Failed to append MsgType to Admin message: %s	Unable to add the MsgType field to an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF3028E	<%s> Failed to append MsgId/Guid to Admin message: %s	Unable to add the MsgId/Guid field to an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF3029E	<%s> Failed to append Version to Admin message: %s	Unable to add the Version field to an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF3030E	<%s> Failed to append bool flag to Admin message: %s	Unable to add the bool flag field to an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF3031E	<%s> Failed to append NumRetries to Admin message: %s	Unable to add the NumRetries field to an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF3032E	<%s> Failed to set RV Admin send subject: %s	An error was returned when the Adapter attempted to set the ECM Admin Subject name in a message to be published. Ensure that a valid subject name is specified in the configuration file.
SXF3033E	<%s> Failed to set RV Admin reply subject: %s	An error was returned when the Adapter attempted to set the ECM Admin Reply Subject name in a message to be published. Ensure that a valid subject name is specified in the configuration file.
SXF3034E	<%s> Failed to send RV Admin msg: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.

Number	Message	Description
SXF3035E	<%s> Failed to subscribe to Admin Request subject <%s> cause=%s	An error was returned when the Adapter attempted to subscribe to an ECM Admin Subject name. Ensure that a valid subject name is specified in the configuration file. Check that TIBCO Rendezvous is available.
SXF3036E	<%s> Failed to subscribe to Admin Reply subject <%s> cause=%s	An error was returned when the Adapter attempted to subscribe to an ECM Admin Subject Reply name. Ensure that a valid subject name is specified in the configuration file. Check that TIBCO Rendezvous is available.
SXF3037E	NULL fileSender in Admin timer callback	A error was detected in the ECM Admin Timer callback support. Check that TIBCO Rendezvous is available. If so, and the problem persists, contact TIBCO technical support.
SXF3038E	<%s> Failed to send Admin msg: cause=%s subject=%s	FilePublisher was unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF3040E	File <%s> can't be deleted. UserId=<%s> does not have ALTER access privileges on the file.	A "removeFile=true" was specified in a FileType definition, but when a Trigger request came in for that File, the requesting user is not authorized to delete the field.
SXF3042W	<%s> will not be deleted when Publishing complete. UserId=<%s> does not have ALTER access privileges on the file	A "removeFile=true" was specified in a FileType definition, but when a Trigger request came in for that File, the requesting user is not authorized to delete the field.

SXF4000 – SXF4999

Number	Message	Description
SXF4002E	API %s failed with exception %s	A TIBCO API failed. Note the return code and contact your TIBCO administrator.

Number	Message	Description
SXF4003E	Input file <%s> does not exist	A Trigger request was received, and the file it was requesting to publish was not present in the Process Input directory.
SXF4004E	File <%s> is present in process library	A copy of the requested file to be published already exists in the Process file. This means that the file is already being published or recovered, and the duplicate publish request is being rejected.
SXF4005E	API %s failed with exception %s	A TIBCO API failed. Note the return code and contact your TIBCO administrator.
SXF4006E	Library not found: %s	Could not find the library specified. Supply a name of a library that exists on the system.
SXF4007E	Input file <%s> does not exist	The input file specified does not exist. Correct the file name, or the library name.
SXF4008E	File <%s> is present in process library.	The specified file already exists in the process library. Move the specified file out of the process library to prevent the Adapter from overwriting the file.
SXF4502E	Generic error: %s %d %d %d %d	File Adapter is reporting an error it encountered. The %s details the cause of the error.
SXF4503E	Generic error: %s %s %d %d %d	File Adapter is reporting an error it encountered. The %s details the cause of the error.
SXF4504W	Library not found: %s	A library needed by the FileAdapter is missing. Re-check the STEPLIB definitions used in the File Adapter JCL.
SXF4505E	Failed to open Pub log file <%s>: cause %d %s. File logging turned off...	The File Adapter was unable to open the trace log file. The second %s details the cause of the problem.
SXF4506E	Error renaming Pub log file <%s> to <%s>: cause %d %s. File logging turned off...	The File Adapter was unable to open the trace log file. The second %s details the cause of the problem.

Number	Message	Description
SXF4507E	Failed to open Pub log file <%s>: cause %d %s. File logging turned off...	The File Adapter was unable to open the trace log file. The second %s details the cause of the problem.

Subscriber Error Messages

SXF5000 – SXF5999

Number	Message	Description
SXF5197E	Config parse error near line %d, failed to parse value	The format or value for a name/tag is not correct. A supplied value is incomplete; typically the value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure all values in the configuration file have matching quotes and braces, and that there are equal signs where required. Check the name/value tags in the section defined in the configuration file. Refer to the manual for the proper format.
SXF5198E	Invalid value set for %s	The format or value for a name/tag is not correct. A supplied value is incomplete; typically the value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure all values in the configuration file have matching quotes and braces, and that there are equal signs where required. Check the name/value tags in the section defined in the configuration file. Refer to the manual for the proper format.
SXF5202E	Unknown/Invalid configuration parameter <%s> at line %d. Ignored	In the FileSubscriber configuration file, a token is invalid in the FileType section. The invalid token is displayed. It could either be a misspelled option name, or an obsolete keyword. Change the configuration file for the associated FileType, to either correct the spelling of the option, or to remove an obsolete or unsupported option.
SXF5203E	Invalid file type: prefix more than 8 char	The file type value supplied must not be more than 8 characters. Change the value supplied to the proper length.

Number	Message	Description
SXF5207E	Block Mode Transfer requires that you specify an genFileSubjectName.	You specified blockTransferMode without specifying a genFileSubjectName. For standard block mode or record-mode ECM, specify a genFileSubjectName.
SXF5208E	Unsupported data set type <%s>	A file type other than SEQ was entered in the configuration file. Correct the file type entry, use only SEQ. Value must be in upper case.
SXF5211E	Process directory [%s] cannot be the same as the output directory.	The process library name is the same as the output library. Use a unique name for the process and the output libraries.
SXF5212E	Invalid output library name: <%s>	The output library name is not a valid file name. Correct the name.
SXF5213E	Invalid output library name: <%s>	The output library name specified is invalid. Use a valid library name. Check for invalid characters. Maximum length is 10 characters.
SXF5214E	Invalid process library name: <%s>	The process library name specified is invalid. Use a valid library name. Check for invalid characters. Maximum length is 10 characters.
SXF5215E	non-Block Mode ECM FileSubscriber must have a file output method	A "standard" block-mode transfer that does not use ECM must specify a AutoGenerate file count or subject name. Supply a valid AutoGenerate parameter in the FileType section of the configuration file.
SXF5216E	Missing <%s> tag	The configuration file is missing a required constraint parameter. The error message indicates which parameter is missing. Check the configuration file.
SXF5217E	Do not use both <position> and <fieldStart> tags)	The configuration file uses both a position and a fieldStart tag. Use only one of the tags to specify a field location.
SXF5218E	Do not use both <position> and <fieldStart> tags)	The configuration file uses both a position and a fieldStart tag. Use only one of the tags to specify a field location.

Number	Message	Description
SXF5219E	Config parse error near line %d, invalid precision: <%s>	The value for the precision tag is not correct. Check the precision tag in the configuration file. Refer to the manual for the proper format.
SXF5220E	Config parse error near line %d, invalid precision: <%s>	The value for the precision tag is not correct. Check the precision tag in the configuration file. Refer to the manual for the proper format.
SXF5221E	Invalid data type <%s> near line <%d>	A invalid field type was specified. Specify a valid field type for conversion to numeric.
SXF5222E	Config parse error near line %d, bad line token: '%s'	In the configuration file for FileSubscriber, a token is invalid in the FILE_LINE element of the FileType section. The invalid token is displayed. Correct or delete the invalid token.
SXF5223E	Field <%s> in <%s %s> must have a position definition...	A position tag was not supplied. Check the configuration file for the field identified, and add a position tag.
SXF5224E	Field <%s> in <%s %s> must have a precision definition for %s conversion...	A precision tag was not supplied. Check the configuration file for the field identified, and add a precision tag.
SXF5225E	Field <%s> in <%s %s> has invalid convertTo value: <%s>...	An invalid conversion value was specified. Only "true" or "false" may be specified for the convertTo tag.
SXF5226E	Field <%s> in <%s %s> has invalid input type: <%s>...	A invalid field type was specified. Specify a valid field type for conversion to numeric.
SXF5227E	Field <%s> in <%s %s> is defined as %s, but FileType must be binary...	The FileType must be binary. Add an isBinary="true" parameter to the file type.
SXF5228E	Config parse error near line %d, failed to parse value	A supplied value is incomplete. Typically the supplied value is missing an opening or closing quote "), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.

Number	Message	Description
SXF5229E	Config parse error near line %d, bad line token: '%s'	In the configuration file for FileSubscriber, a token is invalid in the FILE_LINE element of the FileType section. The invalid token is displayed. Correct or delete the invalid token.
SXF5230E	Config parse error near line %d, failed to parse value	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF5231E	Config parse error near line %d, failed to parse value	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF5232E	Config parse error near line %d, invalid option name: '%s'	A value in the Options section may be misspelled, or may not have the proper case. The invalid option is displayed. Check the spelling of the values in the Options section of the configuration file. Values must be in upper case.
SXFF5233E	Config parse error near line %d, failed to parse options line	In the configuration file, the Options section must have matching opening and closing '}'. Check the configuration file and make sure that the Options section has matching opening and closing '}'.
SXF5234E	Config parse error near line %d, failed to parse value	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.

Number	Message	Description
SXF5235E	Config parse error near line %d, failed to parse value	A supplied value is incomplete. Typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF5236E	Config parse error near line %d, invalid option name: '%s'	A value in the Options section may be misspelled, or may not have the proper case. The invalid option is displayed. Check the spelling of the values in the Options section of the configuration file. Values must be in upper case.
SXF5237E	Config parse error near line %d, failed to parse file type line	In the configuration file, the FileType does not have the proper framing, or is missing required parameters. Check the FileType section of the configuration file and make sure that the section is complete, and that there are matching opening and closing braces ({ }).
SXF5238E	Config parse error near line %d, invalid option name: '%s'	A value in the Options section may be misspelled, or may not have the proper case. The invalid option is displayed. Check the spelling of the values in the Options section of the configuration file. Values must be in upper case.
SXFF5239E	Config parse error near line %d, failed to parse options line	A supplied value is incomplete, typically the supplied value is missing an opening or closing quote ("), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF5240E	Only %d file types are supported	You have specified too many File Types in the configuration file. Reduce the number of File Type definitions

Number	Message	Description
SXF5241E	Config parse error near line %d, invalid section name: '%s'	A section name in the configuration file is incorrect. The section name must be one of Trace, Options, or FileType. Check the spelling of the section names of the configuration file. Names are case sensitive.
SXF5242E	Config parse error near line %d, failed to find ']'	In the configuration file, a section description (Trace, Options, or FileType) does not have a closing ']'. Check the configuration file and make sure that all section descriptions are formed correctly.
SXF5243E	Config parse error near line %d, failed to find section'	A section name in the configuration file is missing. The section name must be one of Trace, Options, or FileType. Check the spelling of the section names in the configuration file. Names are case sensitive.
SXF5244E	Config parse error near line %d, failed to terminate expression	A section was not properly configured in the configuration file. Check the configuration file and correct the error.
SXF5245E	Missing [%s] section	A section name in the configuration file is missing. The section name must be one of Trace, Options, or FileType. Check the spelling of the section names in the configuration file. Names are case sensitive.
SXF5246E	Missing [%s] section	A section name in the configuration file is missing. The section name must be one of Trace, Options, or FileType. Check the spelling of the section names in the configuration file. Names are case sensitive.
SXF5247E	Missing [%s] section	A section name in the configuration file is missing. The section name must be one of Trace, Options, or FileType. Check the spelling of the section names in the configuration file. Names are case sensitive.
SXF5248E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.

Number	Message	Description
SXF5249E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.
SXF5250E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.
SXF5251E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.
SXF5252E	Missing [%s] section option: %s	The configuration file is missing the section option specified. You must specify the indicated option. Add the option to the configuration file.
SXF5253E	Missing %s definitions for file prefix <%s>	An input library has not been specified for this file type. Correct the configuration file.
SXF5254E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF5256E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.
SXF5257E	Library not found: %s	The specified library was not found. Make sure that the specified library exists and is available.
SXF5258E	Library not found: %s	The specified library was not found. Make sure that the specified library exists and is available.
SXF5259E	Missing %s definition for file prefix <%s>	The specified section for this file type is missing. Add the definitions to the configuration file.

Number	Message	Description
SXF5260E	Library not found: %s	The specified library was not found. Make sure that the specified library exists and is available.
SXF5261E	Library not found: %s	The specified library was not found. Make sure that the specified library exists and is available.
SXF5262E	Failed to open config file: %s	Unable to open the configuration file or the configuration file was not found. Make sure that the configuration file exists, and that it can be found by the Adapter.
SXF5264E	Failed to open TIB/Rendezvous RV. cause=%s	Unable to start TIBCO Rendezvous, it may be inactive. Check with your TIBCO administrator that TIBCO Rendezvous is available.
SXF5265E	Failed to create RV Transport session: service=%s network=%s daemon=%s. EMS/RV Error=%s	Unable to start TIBCO Rendezvous, it may be inactive. Check with your TIBCO administrator that TIBCO Rendezvous is available.
SXF5266E	Failed to open TIBCO Rendezvous CM -- %s	Unable to start TIBCO Rendezvous, it may be inactive. Check with your TIBCO administrator.
SXF5267E	Failed to create RV transport: %s	Unable to create a network transport. TIBCO Rendezvous may be inactive. Check with your TIBCO administrator.
SXF5268E	Failed to create RVCM Transport: %s	Unable to create a network transport. TIBCO Rendezvous may be inactive. Check with your TIBCO administrator.
SXF5269E	Configuration File Error near line <%d> missing ',' .	There is an error in the configuration file. Verify that the configuration file is coded properly.
SXF5270E	Found definitions for both Reliable and Certified mode	The configuration file contains definitions for both modes. The modes are mutually exclusive. Specify only one mode in the configuration file.

Number	Message	Description
SXF5271E	Found definitions for both Reliable and Certified mode	The configuration file contains definitions for both modes. The modes are mutually exclusive. Specify only one mode in the configuration file.
SXF5272E	Missing tag <%s> for %s	The configuration file is missing a required parameter. The error message indicates which parameter is missing. Check the configuration file and correct the discrepancy.
SXF5273E	Missing tag <%s> for %s	The configuration file is missing a required parameter. The error message indicates which parameter is missing. Check the configuration file and correct the discrepancy.
SXF5274E	Cannot specify both vsamReplaceRecords and vsamUpsertRecords =true	The FileType definition has both parameters specified. This is not allowed. Choose one or the other.
SXF5276E	Memory allocation error of LineField struct. Length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF5277E	lineBuff Option Memory allocation failed. Length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF5278E	optionvalue Memory allocation failed. Length=%d	An attempt to allocate memory failed. It normally indicates insufficient memory.
SXF5279E	optionvalue Memory allocation failed. Length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF5280E	lineBuff Option Memory allocation failed. Length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF5281E	lineBuff Option Memory allocation failed. Length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF5282E	lineBuff Option Memory allocation failed. Length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF5283E	Missing %s definition for file prefix <%s>	A FileType requires a Prefix definition, and none was found.

Number	Message	Description
SXF5284E	Missing %s definition for file prefix <%s>	A FileType requires a Prefix definition, and none was found.
SXF5286E	Missing %s definition for file prefix <%s>	A FileType requires a Prefix definition, and none was found.
SXF5287E	Missing %s definition	No Adapter name has been specified. Add the definition(s) to the configuration file.
SXF5288E	FileOptions Memory allocation failed	An attempt to allocate memory failed. It normally indicates insufficient memory.
SXF5289E	Config parse error near line %d, failed to find ']'	There is an error in the configuration file. Verify that the configuration file is coded properly.
SXF5291E	Invalid file type: no prefix	A FileType requires a Prefix definition, and none was found.
SXF5292W	Caution: Binary files must always be Fixed Block. usedFixedRecordFile=false parameter is ignored.	The Adapter requires that binary files must use fixed block (FB) format. Correct the configuration file.
SXF5293E	There is a config error. reading till next section is encountered :[%d]	A supplied value is incomplete. Typically the supplied value is missing an opening or closing quote "), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF5294E	There is an error while parsing the name value pairs. exit parsing	A supplied value is incomplete. Typically the supplied value is missing an opening or closing quote "), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.

Number	Message	Description
SXF5295E	There is an configuration error at line <%d>:[%s]	A supplied value is incomplete. Typically the supplied value is missing an opening or closing quote "), brace ({ }), or an equal sign (=). Make sure that the values in the configuration file have matching quotes and braces, and that there are equal signs where required.
SXF5296E	blkSizeAlloc <%d>, is not a multiple of lineLength [%d] for fixed block	The value specified for blkSizeAlloc= is not a multiple of the record length (lineLength). Recheck the RECL and BLKZSIZE attributes of the file on your system, and set blkSizeAlloc to a valid multiple of the RECL.
SXF5297E	blkSizeAlloc <%d>, is not a multiple of lineLength [%d] for variable block	The value specified for blkSizeAlloc= is not a correct multiple of the record length (lineLength). It must be (RECL * n) + 4. Re-check the RECL and BLKZSIZE attributes of the file on your system, and set blkSizeAlloc to a valid multiple+4 of the RECL.
SXF5299W	Target file <%.48s> not found. Requested by user=%s on Subject/Destination=%s.	The File Adapter cannot find a valid entry for this file. It is either mis-spelled or has been deleted.
SXF5302W	Both Code Pages must be configured. Single code page config is ignored.	Normally, both HOSTCODEPAGE and NETWORKCODEPAGE should be specified when overriding the default code pages used by the File Adapter.
SXF5303W	Code Page Setup failed: reason=<%s>. Will use defaults instead.	Unable set up the requested code pages. Verify that the code page settings are valid for your system and that TIBCO Rendezvous is active. Check with your TIBCO administrator.
SXF5306E	Config ERROR: you have defined both record-mode ECM and Block-Mode ECM parms in the same Filetype. Configuration has been overriden to use Block Mode ECM.	You specified parameters for both Block Mode ECM and record-mode ECM versions of ECM block mode. Determine which version of the ECM protocol you, then remove the parameters that apply to the other version of the protocol.

Number	Message	Description
SXF5307E	Config ERROR: you have defined both forcePublishedFileName and generateFileOnNumberOfMessages=%d parms. This combination is not allowed.	The file type has been defined with both forcePublishedFileName and generateFileOnNumberOfMessages parameters. Remove one of the parameters from the configuration file.
SXF5323E	Config ERROR: you must define retryInterval > 0 if Locked File retry is enabled with noOfRetries > 0	There is an error in the configuration file. Verify that the configuration file is coded properly.
SXF5326E	Invalid %s setting=%s. Must be RecordMode, RecordModeECM, BlockModeSFT, or BlockModeECM	Valid values for the transferType parameter are RecordMode, RecordModeECM, BlockModeSFT, or BlockModeECM.

SXF6000 – SXF6999

Number	Message	Description
SXF6013E	Error %d on timer create: %s	Unable to create the RV heartbeat callback timer.
SXF6019W	Received Signal, exiting app...	Cancel or error signal received; exiting application.
SXF6020W	Could not remove SIGINT signal handler: %s	The File Adapter encountered an error while terminating. If the problem persists, contact TIBCO technical support.
SXF6021W	Could not remove SIGTERM signal handler: %s	The File Adapter encountered an error while terminating. If the problem persists, contact TIBCO technical support.
SXF6022W	Could not setup SIGINT signal handler: cause=%s	The File Adapter encountered an error while terminating. If the problem persists, contact TIBCO technical support.

Number	Message	Description
SXF6023W	Could not setup SIGTERM signal handler: cause=%s	The File Adapter encountered an error while terminating. If the problem persists, contact TIBCO technical support.
SXF6024E	Error in sending Heartbeat message: cause=%s Subject/Destination=%s	An error occurred when sending a TIBCO heartbeat message. The Adapter may not be active. Contact your TIBCO administrator.
SXF6025E	Error in sending Heartbeat message: cause=%s Subject/Destination=%s	An error occurred when sending a TIBCO heartbeat message. The Adapter may not be active. Contact your TIBCO administrator.
SXF6026E	Failed to unsubscribe for exit Subject/Destination <%s>: cause=%s	The File Adapter encountered an error while terminating. If the problem persists, contact TIBCO technical support.
SXF6027E	Failed to unsubscribe for refresh options Subject/Destination <%s>: %s	The File Adapter encountered an error while terminating. If the problem persists, contact TIBCO technical support.
SXF6030E	Heartbeat timer removal failed: cause=%s	The File Adapter encountered an error while terminating. If the problem persists, contact TIBCO technical support.
SXF6031E	Failed to send error message: cause=<%s> Subject/Destination=%s	File Adapter was unable to send a message to the network. The %s cause details why. It may indicate a network or RV daemon problem.
SXF6032E	FileHandler Global-List Memory allocation failed	An attempt to allocate memory failed. It normally indicates insufficient memory.
SXF6033E	Failed to subscribe to Subject/Destination <%s> cause=%s	Unable to create a listener. TIBCO Rendezvous may be unavailable. Check with your TIBCO administrator.
SXF6034E	FileHandler Memory allocation failed for new [FileType]. Length=%d	An attempt to allocate memory failed. It normally indicates insufficient memory.
SXF6035E	Memory allocation failed for SaveTimer entry. Length=%d	An attempt to allocate memory failed. It normally indicates insufficient memory.
SXF6036E	HeartBeat Msg Memory allocation failed. Length=%d	An attempt to allocate memory failed. It normally indicates insufficient memory.

Number	Message	Description
SXF6037E	HeartBeat Nested Msg Memory allocation failed. Length=%d	An attempt to allocate memory failed. It normally indicates insufficient memory.
SXF6038E	Failed to create Heartbeat message: cause=%s	FileSubscriber was unable to create a heartbeat message. Check that TIBCO Rendezvous is available.
SXF6039E	Failed to add [%s] to Heartbeat message: %s	Unable to add the specified string to a heartbeat message. Check that TIBCO Rendezvous is available.
SXF6040E	Failed to add [%s] to Heartbeat message: %s	Unable to add the specified string to a heartbeat message. Check that TIBCO Rendezvous is available.
SXF6041E	Failed to add file prefix for: %d:%s cause=%s	Unable to add the file prefix field to a heartbeat message. Check that TIBCO Rendezvous is available.
SXF6042E	Failed to add file extension for %d:%s cause=%s	Unable to add the file extension to a heartbeat message. Check that TIBCO Rendezvous is available.
SXF6043E	Failed to add status for: %d:%s cause=%s	Unable to add the status string to the heartbeat timer message. Check with the TIBCO administrator.
SXF6044E	Failed to add Begin Time for: %d:%s cause=%s	Unable to add the Begin Time string to the heartbeat timer message. Check with the TIBCO administrator.
SXF6045E	Failed to add NumMsgs for: %d:%s cause=%s	Unable to add the NumMsgs string to the heartbeat timer message. Check with the TIBCO administrator.
SXF6046E	Failed to add BlockNum for: %d:%s cause=%s	Unable to add the BlockNum string to the heartbeat timer message. Check with the TIBCO administrator.
SXF6047E	Failed to add LineNum for: %d:%s cause=%s	Unable to add the LineNum string to the heartbeat timer message. Check with the TIBCO administrator.

Number	Message	Description
SXF6048E	Aborting heartbeat message publish	Too many previous AddMessage errors forced the FileSubscriber to abandon sending a heartbeat timer message. Check with the TIBCO administrator.
SXF6049E	Failed to set Heartbeat send Subject/Destination: %s cause=%s	An error was returned when the Adapter attempted to set the subject/destination name in a Heartbeat message to be published. Ensure that a valid subject/destination name is specified in the configuration file.
SXF6050E	Failed to destroy message.	Unable to destroy a published heartbeat message. Make sure that TIBCO Rendezvous is available.
SXF6051E	Failed to create error message: cause=%s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF6052E	Failed to add [%s] to error message: %s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF6053E	Failed to set error send Subject/Destination: %s cause=%s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF6054E	Failed to send error message: cause=<%s> Subject/Destination=%s	File Adapter was unable to send a message to the network. The %s cause details why. It may indicate a network or RV daemon problem.
SXF6055E	Failed to destroy error message: reason=%s	File Adapter encountered an error when destroying. If it persists, contact TIBCO technical support.
SXF6056E	Configuration file from -config parameter and QINLINE file not found. Will use default config filename <%s> instead.	No configuration file was specified. This is probably an error by the operator. Double-check that you really want to use the default configuration file.
SXF6057E	Failed to get hostname from system. Error code returned by 'gethostname' is = %d	Unable to extract the host name being used by TCPIP. Probably indicates that TCP/IP is not up or not functioning.

Number	Message	Description
SXF6058E	Could not find a usable Sockets connection.	A missing TCP Steplib entry is causing problems.
SXF6059E	Failed to get hostname from system. Error code returned by 'gethostname' is = %d	Unable to extract the host name being used by TCPIP. Probably indicates that TCP/IP is not up or not functioning.
SXF6061E	Unable to open security mapping file: %s cause=%d %s.	The security mapping file may be mis-named or under RACF protection.
SXF6062E	Security mapping file %s has invalid data: Line=%d <%s>	There is an error in the security mapping file. Verify that the configuration file is coded properly.
SXF6063E	Memory allocation failed while processing Security mapping file %s at Line=%d	An attempt to allocate memory failed. It normally indicates insufficient memory for the REGION. If increasing the REGION size does not solve the problem, then contact TIBCO technical support.
SXF6065E	Failed to create the Default Data RV queue-- reason: %s	TIBCO Rendezvous may be unavailable or may not be active. Check with your TIBCO administrator.
SXF6066E	Failed to set the RV limit policy-- reason: %s	TIBCO Rendezvous may be unavailable or may not be active. Check with your TIBCO administrator.
SXF6067E	Failed to subscribe to Subject/Destination <%s> reason: %s	TIBCO Rendezvous may be unavailable or may not be active. Check with your TIBCO administrator.
SXF6068E	Failed to create the Priority Admin RV queue-- reason: %s	TIBCO Rendezvous may be unavailable or may not be active. Check with your TIBCO administrator.
SXF6069E	Failed to set the RV Queue Priority to %d for %s. Reason: %s	Setting RV Queue Priority via tibrvQueue_SetPriority API has failed.

SXF7000 – SXF7999

Number	Message	Description
SXF7001E	<%s> TEXT fopen failed. cause=%d: %s	The open of a TEXT file failed. The cause contains the reason for the failure.
SXF7002E	<%s> fclose failed. cause=%d: %s	The close of a file failed. The cause contains the reason for the failure.
SXF7003E	<%s> BINARY fopen failed. cause=%d: %s	The open of a BINARY file failed. The cause contains the reason for the failure.
SXF7004E	<%s> fclose failed. cause=%d: %s	The close of a file failed. The cause contains the reason for the failure.
SXF7005E	<%s> Generating file <%s> due to loss of data	A target file is being generated, but data has been lost or truncated. This indicates either an out of space error, or a configuration (incorrect file size) error.
SXF7006E	<%s> Exiting because of file save error. Deleted PRP and PRG	A target file could not be properly saved and closed. The File Adapter was configured to shutdown in such a case.
SXF7026E	<%s> close failed during trailer append. errno=%d errtext=%s	The close of a file failed after file trailer data was added. The cause contains the reason for the failure.
SXF7032E	<%s> close failed during header append. errno=%d errtext=%s	The close of a file failed after file header data was added. The cause contains the reason for the failure.
SXF7035W	<%s> Recovery mode: Generating error file=<%s> due to %s	The File Adapter is performing recovery after a shutdown or crash, and is generating an ERR file because of an error encountered.
SXF7036W	<%s> File transfer: Generating error file=<%s> due to %s	The File Adapter is was generating a file but received an error, so it is converting it to an ERR file.
SXF7069E	<%s> Recovery SEQ fopen failed using file=<%s> mode='r'. cause=%d: %s	The close of a file failed. The cause contains the reason for the failure.

Number	Message	Description
SXF7070W	<%s> SEQ fclose failed during recovery. cause=%d: %s	The close of a file failed during Recovery/Restart processing. The cause contains the reason for the failure.
SXF7088W	<%s> Truncating TEXT data - discarded %d bytes. expectnl=%d	The data received for a FileType exceeds the max length configured for the File Type. The data has been truncated.
SXF7089W	<%s> Truncating TEXT data - discarded %d bytes. expectnl=%d	The data received for a FileType exceeds the max length configured for the File Type. The data has been truncated.
SXF7102W	Retry timer pop: retry # %d trying to move .CWK to target file=<%s>. (timer interval=%d secs)	A target file is temporarily locked by another job. This indicates when the File Adapter attempts to again try to see if the file is now free.
SXF7103E	Move operation could not remove file <%s>. Deleting PRP and PRG	Could not remove the specified file. The transfer is being aborted. See the previous error message in the trace log for the cause of the problem.
SXF7104W	<%s> ### Continuing to accept data in working file	New data has been received into the CWK file after a previous error (e.g. failed Generate). The File Adapter will continue to try to process the incoming data.
SXF7106W	<%s> Retry limit=<%d> for Target File Locked <%s> has been reached. Deleting PRP and PRG files too.	A target file that was temporarily locked by another job has exceeded its max retry limit. The received new file data will be converted to an ERR file.
SXF7107E	<%s> Exiting because of file save error	A target file could not be properly saved and closed. The File Adapter was configured to shutdown in such a case.
SXF7108W	<%s> ### Continuing to accept data in working file	New data has been received into the CWK file after a previous error (e.g. failed Generate). The File Adapter will continue to try to process the incoming data.
SXF7117W	Parse error: bad variable name <%s>	Encountered an invalid variable name. Correct identified invalid variable name in the configuration file.

Number	Message	Description
SXF7118W	Parse error: missing end of variable	The configuration parser program was unable to properly parse a variable. Check the configuration file for adding header or trailer records. These must be framed with '%'.
SXF7119W	<%s> Move operation could not remove workfile: cause=%d: %s. Caller=%s	Could not remove the specified file. The transfer is being aborted. The cause contains the reason for the error.
SXF7120W	Failed to convert time value, using current time	Unable to convert a TIME value from a received message, and it will use the current system time for the TIME field in an output record. Check the configuration file to ensure that the field identified contains a TIME value.
SXF7121W	Failed to extract nested msg. field <%s>	Unable to construct the subject/destination from the message node. Check the configuration file for this message.
SXF7122W	Failed to extract field <%s> to form Subject/Destination name	Unable to construct the subject/destination from the field data. Check the configuration file for this message.
SXF7123W	Failed to write data from file <%s> - wrote %d out of %d	A file write failed. It is usually due to an out of space error. See the previous error message in the trace log for the exact cause.
SXF7124W	Move operation could not remove file <%s>	Could not remove the specified file. The transfer is being aborted. See the previous error message in the trace log for the cause of the problem.
SXF7125W	Move operation could not remove file <%s>	Could not remove the specified file. The transfer is being aborted. See the previous error message in the trace log for the cause of the problem.
SXF7126W	<%s> Block number %d arrived before IFACTIVE_REPLY Ack, Discarding the block.	A timing race occurred between Block Mode ECM Publisher and Subscriber at startup. The message block will be discarded and re-transmitted once the startup handshake sequence is complete.

Number	Message	Description
SXF7127W	<%s> Dup Block number %d matches previous block (%d), skipping write for workfile	FileSubscriber has received a block of data that has already been received. The block is ignored.
SXF7128W	<%s> Block number %d is not next in sequence. Expected blk=%d	FileSubscriber received a block of data out of order.
SXF7129W	<%s> TEXT file write fixed width Field failed for workfile	A file write failed. It is usually due to an out of space error. See the previous error message in the trace log for the exact cause.
SXF7130W	<%s> TEXT file write padded Field Value failed	A file write failed. It is usually due to an out of space error. See the previous error message in the trace log for the exact cause.
SXF7131W	Failed to convert time value, using current time	Unable to convert a TIME value from a received message, and it will use the current system time for the TIME field in an output record. Check the configuration file to ensure that the field identified contains a TIME value.
SXF7132W	<%s> Workfile not removed. Will be renamed to .ERR	Source file is retained while moving a file. It is possible that the source file was in use by another process when the Adapter attempted to move the file. Remove the source file if it is not needed.
SXF7133W	<%s> Move operation could not remove workfile: cause=%d: %s	Source file is retained while moving a file. It is possible that the source file was in use by another process when the Adapter attempted to move the file. Remove the source file if it is not needed.
SXF7134W	<%s> Move operation could not remove workfile: cause=%d: %s	Source file was in use by another process when the Adapter attempted to remove the file
SXF7135W	<%s> Timer: Move operation could not remove workfile. Deleting PRP and PRG	Source file was in use by another process when the Adapter attempted to remove the file

Number	Message	Description
SXF7136W	Received badly formatted message	A message was received that does not match the configuration file. The configuration file/field specification for a received message does not correspond to the message received. Correct the file/field specification for the subject/destination being subscribed to, or check the source that published the message.
SXF7137W	Received badly formatted message	A message was received that does not match the configuration file. The configuration file/field specification for a received message does not correspond to the message received. Correct the file/field specification for the subject/destination being subscribed to, or check the source that published the message.
SXF7138E	Failed to unsubscribe to Subject/Destination <%s>: cause=%s	During shutdown, the FileSubscriber was unable to terminate the subscription to a Subject. The cause %s specifies what the problem was
SXF7139E	Failed to unsubscribe to Subject/Destination <%s>: cause=%s	During shutdown, the FileSubscriber was unable to terminate the subscription to a Subject. The cause %s specifies what the problem was
SXF7140E	Failed to unsubscribe to Subject/Destination <%s>: cause=%s	During shutdown, the FileSubscriber was unable to terminate the subscription to a Subject. The cause %s specifies what the problem was
SXF7141E	Failed to unsubscribe to Subject/Destination <%s>: cause=%s	During shutdown, the FileSubscriber was unable to terminate the subscription to a Subject. The cause %s specifies what the problem was
SXF7142E	Retry Timer removal failed: cause=%s	During shutdown, the FileSubscriber was unable to terminate RV. The cause %s specifies the problem.
SXF7145E	Out of space condition reached while writing out to file <%s>. Terminating processing for that file.	The I/O to the file has reached an out of space condition. Any subsequent data destined for that file is rejected and discarded.

Number	Message	Description
SXF7146E	Out of space condition reached while writing out to file <%s>. Terminating processing for that file.	The I/O to the file has reached an out of space condition. Any subsequent data destined for that file is rejected and discarded.
SXF7148E	Confirm Message create RVMsg failed	Unable to allocate memory to create a new message. Report this error to your TIBCO administrator.
SXF7149E	Compression block Memory reallocation failed. Length=%d	An attempt to re-allocate memory failed. It normally indicates insufficient memory.
SXF7150E	<%s> Error in sending RVCM message: cause=%s Subject/Destination=%s	An error occurred when sending an RVCM message. The Adapter may not be active. Contact your TIBCO administrator.
SXF7151E	<%s> Error in sending message: cause=%s	An error occurred when sending a message. The Adapter may not be active. Contact your TIBCO administrator.
SXF7152E	NULL FileHandler in retry timer callback	This is an internal error. Contact TIBCO technical support.
SXF7153E	Unable to extract CM message seqnum: %s	This is an internal error. Contact TIBCO technical support.
SXF7154E	Unable to extract publisher name: %s	This is an internal error. Contact TIBCO technical support.
SXF7155E	Failed to subscribe to Subject/Destination <%s>: cause=%s	Unable to create a listener. TIBCO Rendezvous may be unavailable. Check with your TIBCO administrator.
SXF7156E	Failed to subscribe to Subject/Destination <%s>: cause=%s	Unable to create a listener. TIBCO Rendezvous may be unavailable. Check with your TIBCO administrator.
SXF7157E	Failed to subscribe to Subject/Destination <%s>: cause=%s	Unable to create a listener. TIBCO Rendezvous may be unavailable. Check with your TIBCO administrator.
SXF7158E	Failed to subscribe to Subject/Destination <%s>: cause=%s	Unable to create a listener. TIBCO Rendezvous may be unavailable. Check with your TIBCO administrator.

Number	Message	Description
SXF7159E	<%s> Error moving workfile to target file=%s. Reason=%s. errno=%d file_rc=%d secur_rc=%d append=%d eof=%d	A target file could not be properly saved and closed. The reason code describes what the problem was.
SXF7167E	<%s> Failed to open workfile: errno=%d err text=%s	The open of a file failed. The errtext contains the reason for the failure.
SXF7168E	<%s> Failed to malloc TEXT LRECL buffer of length=%d	An out of memory error was reached.
SXF7170E	<%s> Failed to open Progress file: <%s> errno=%d errtext=%s	The open of a FileAdapter PRG progress file failed. The errtext contains the reason for the failure.
SXF7172E	<%s> Command <%s> returned error rc=%d	The command or program specified for executeAfterProcess or executeBeforeProcess did not execute properly.
SXF7174E	<%s> Command <%s> returned error rc=%d	The command identified did not complete successfully. Check the command specified for executeBeforeProcess or executeAfterProcess and ensure that the commands are valid.
SXF7180E	<%s> Error moving workfile to target file=%s	Unable to move a file. Either the target data set is not available or is invalid. Provide a valid data set name in the configuration file.
SXF7181E	<%s> Error moving workfile to target file=%s	Unable to move a file. Either the target data set is not available or is invalid. Provide a valid data set name in the configuration file.
SXF7182E	<%s> Error moving workfile to target file=%s	Unable to move a file. Either the target data set is not available or is invalid. Provide a valid data set name in the configuration file.
SXF7183E	<%s> Failed to close .PRG file=%s, aborting genOutFile	Unable to close the progress file. Check the processlibrary. The file may be corrupted.
SXF7184E	<%s> Failed to remove .PRG file=%s, aborting genOutFile	Unable to remove the progress file. Check the process library. The file may be corrupted or not properly cataloged.

Number	Message	Description
SXF7185E	<%s> Failed to remove .PRG file=%s, aborting genOutFile	Unable to remove the progress file. Check the process library. The file may be corrupted or not properly cataloged.
SXF7186E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and ensure that the configuration file matches the received message.
SXF7187E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and ensure that the configuration file matches the received message.
SXF7188E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and be certain that the configuration file matches the received message.
SXF7189E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and ensure that the configuration file matches the received message.
SXF7190E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and ensure that the configuration file matches the received message.
SXF7191E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and ensure that the configuration file matches the received message.
SXF7192E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and ensure that the configuration file matches the received message.
SXF7193E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and be certain that the configuration file matches the received message.

Number	Message	Description
SXF7194E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file and ensure that the configuration file matches the received message.
SXF7195E	<%s> default Type mismatch while retrieving value: %s, pf_type: %d	The field type received does not match the expected data type that was configured. Correct the configuration file and ensure that the configuration file matches the received message.
SXF7196E	<%s> Failed to get RV field: cause=%s	Unable to extract a requested field from the message node. Check the configuration file for this message to ensure the fields in the message are correctly specified.
SXF7197E	Log Write <%s> failed: errno=%d last_op=%d	The write operation to the log file has failed. This message gives the error codes and the text message of the error codes.
SXF7198E	<%s> LOG file <%s> reached out of space. Wrapping to Begin of file.	The log file is complete and the vsamUseLog parameter is not set. After this message, the Adapter starts writing in the log from the beginning.
SXF7199E	<%s> Output file failed: errno=%d last_op=%d	The write operation to the target file has failed. This message gives the error codes and the text message of the error codes.
SXF7200E	<%s> Text File Write failed: errno=%d err text=%s	The write operation to the TEXT file has failed. This message gives the error codes and the text message of the error codes.
SXF7201E	<%s> Binary File Write failed: errno=%d last_op=%d	The write operation to the BINARY file has failed. This message gives the error codes and the text message of the error codes.
SXF7204E	<%s> Failed to open Work file. cause=%d: %s	Unable to open a CWK work file. It usually indicates an out of space problem, or a RACF authorization problem. The cause contains the exact cause of the error.

Number	Message	Description
SXF7205E	<%s> Failed to open Work file. cause=%d: %s	Unable to open a CWK work file. It usually indicates an out of space problem, or a RACF authorization problem. The cause contains the exact cause of the error.
SXF7206E	<%s> Failed to close .PRG file=%s, aborting recovery	A close on a Progress PRG file being used failed. Save the %s reason code and discuss the issue with your systems programmer.
SXF7207E	<%s> Failed to malloc TEXT LRECL buffer of length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF7208E	<%s> Unable to open .PRP file=%s, aborting recovery	Unable to open a CWK work file. It usually indicates an out of space problem, or a RACF authorization problem. The cause contains the exact cause of the error.
SXF7209E	<%s> Failed to read data in PRP file, aborting recovery	Unable to read data cached in the partial record file. Check the process directory. The file may be corrupted.
SXF7210E	<%s> Failed to read data in progress file, aborting recovery	Unable to read the progress file. Check the process directory. The file may be corrupted.
SXF7211E	<%s> Failed to remove .PRG file=%s, aborting recovery	Unable to remove a file. Either the target data set is not available or is invalid. Report this error to your TIBCO administrator.
SXF7212E	<%s> Failed to recover Work file=<%s>. fopen with Append failed. Cause=%d: %s	Unable to recover a work file. The file may be corrupted. Contact the TIBCO administrator.
SXF7213E	<%s> Failed to allocate memory for TEXT LRECL buffer of length=%d	Unable to allocate memory. Report error to your TIBCO administrator.
SXF7214E	<%s> No hostname supplied on message with Security check	Security is turned on for the Adapter, but an incoming message did not have a Publisher host name associated with it. The request is discarded. Verify that the Publisher is set up to automatically send the Publisher's host name to the Subscriber.

Number	Message	Description
SXF7215E	<%s> No User Id supplied on message with Security check	Security is turned on for the Adapter, but an incoming message did not have a User Id associated with it. The request is discarded. Verify that the Publisher is setup to automatically send the UserId to the Subscriber.
SXF7216E	<%s> Failed to extract checksum from end-of-file message	Invalid configuration was specified.
SXF7217E	Failed to write data from file <%s> - wrote %d out of %d	Could not add the data to the file, probably due to an out of space issue. Check the system log for related space problems.
SXF7218E	<%s> Failed to retrieve sequence/block_id number from ECM msg. cause=%s on Subject/Destination=<%s>. field=%s	Unable to retrieve a block number from a block transfer file. Ensure that the file being subscriber to is being published in block mode.
SXF7219E	<%s> Failed to GUID from msg. cause=%s	Unable to retrieve unique GUID from a block transfer file. Ensure that the file being subscriber to is being published in block transfer mode.
SXF7220E	<%s> Failed to retrieve block DATA for block # %d for workfile. cause=%s	Unable to retrieve data element from a block transfer file. Ensure that the file being subscriber to is being published in block transfer mode.
SXF7221E	<%s> Write Failed for block num=%d on workfile. cause=%d: %s	Could not add the new block of data to the file, probably due to an out of space issue. Check the system log for related space problems.
SXF7222E	<%s> TEXT file write delimiter %s failed for workfile	Could not add the delimiter data to the file, probably due to an out of space issue. Check the system log for related space problems.
SXF7223E	<%s> Failed to retrieve field <%s> from message for workfile	Unable to retrieve field, possible configuration mismatch. Check the configuration file, and ensure that it matches the received message

Number	Message	Description
SXF7224E	<%s> TEXT file write of delimited Value <%s> failed	Could not add the delimited value data to the file, probably due to an out of space issue. Check the system log for related space problems.
SXF7225E	<%s> TEXT file write of trail padding <%s> failed	Could not add the padding data to the file, probably due to an out of space issue. Check the system log for related space problems.
SXF7226E	<%s> Failed to retrieve field <%s> from message for workfile	Unable to retrieve field, possible configuration mismatch. Check the configuration file, and ensure that it matches the received message.
SXF7227E	<%s> TEXT file write fixed width Field <%s> failed	Could not add the value data to the file, probably due to an out of space issue. Check the system log for related space problems.
SXF7228E	<%s> TEXT file write padded Field Value failed: <%s>	Could not add the value data to the file, probably due to an out of space issue. Check the system log for related space problems.
SXF7229E	<%s> TEXT file fflush failed	I/O error received for fflush operation. Check previous message in trace log from more details on the cause.
SXF7231E	<%s> Failed to retrieve field <%s> from message for workfile	Failed to retrieve field, possible configuration mismatch on the message definition. Check the configuration file, and ensure that it matches the received message.
SXF7232E	<%s> Data type or Data content error on field <%s>	Data value retrieved from file is invalid. Possible configuration mismatch on the message definition.
SXF7233E	<%s> Failed to retrieve field <%s> from message for workfile	Failed to retrieve field, possible configuration mismatch. Check the configuration file, and ensure that it matches the received message.
SXF7234E	<%s> BINARY file write failed <%s>	The write operation to the BINARY file has failed. This message gives the error codes and the text message of the error codes.

Number	Message	Description
SXF7235E	<%s> BINARY file write failed details: errno=%d	The write operation to the BINARY file has failed. This message gives the error codes and the text message of the error codes.
SXF7238E	<%s> Write Failed for block num=%d on workfile for %d TEXT records of max %d bytes each. Error=%d: %s	The write operation to the TEXT file has failed. This message gives the error codes and the text message of the error codes.
SXF7239E	<%s> Unable to open .PRP file=%s, data exposure if crash	The open for a TEXT file's partial data buffer file has failed. This message gives the error codes and the text message of the error codes.
SXF7240E	<%s> Unable to write to .PRP file=%s, data exposure if crash	The write operation to the TEXT file partial data buffer file has failed. This message gives the error codes and the text message of the error codes.
SXF7241E	<%s> Failed to initialize Confirm message	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF7242E	<%s> Failed to append Block Num to Confirm message	Unable to add the BlockNumber field to an ECM Confirm message. Make sure that TIBCO Rendezvous is available.
SXF7243E	<%s> Failed to append Status to Confirm message	Unable to add the Status field to an ECM Confirm message. Make that TIBCO Rendezvous is available.
SXF7244E	<%s> Failed to append Subscriber Name to Confirm message	Unable to add the Subscriber Name field to an ECM Confirm message. Make sure that TIBCO Rendezvous is available.
SXF7245E	<%s> Failed to append GUID to Confirm message	Unable to add the GUID field to an ECM Confirm message. Make sure that TIBCO Rendezvous is available.
SXF7246E	<%s> Failed to append Version to Confirm message	Unable to add the BlockNumber field to an ECM Confirm message. Make sure that TIBCO Rendezvous is available.

Number	Message	Description
SXF7247E	<%s> Failed to set RVCM send target: cause=%s Subject/Destination=%s	An error was returned when the Adapter attempted to publish a message. Ensure that a valid subject/destination name is specified in the configuration file.
SXF7248E	<%s> Failed to set time limit for certified message: %s	Unable to set a time limit for a certified message. Check that TIBCO Rendezvous is available and operating properly.
SXF7249E	<%s> Error in sending RVCM message: cause=%s Subject/Destination=%s	An error occurred when sending an RVCM message. The Adapter may not be active. Contact your TIBCO administrator.
SXF7250E	<%s> Failed to set send target: cause=%s Subject/Destination=%s	An error occurred when sending an RVCM message. The Adapter may not be active. Contact your TIBCO administrator.
SXF7251E	<%s> Error in sending message: cause=%s Subject/Destination=%s	An error occurred when sending an RVCM message. The Adapter may not be active. Contact your TIBCO administrator.
SXF7252E	<%s> Failed to initialize GenFile message: cause=%s	Unable to create a genFilePublish message. Ensure that TIBCO Rendezvous is available.
SXF7253E	<%s> Failed to add filename to genFilePublish message: cause=%s	Unable to add the filename field to a genFilePublish message. Check the configuration file to ensure that a valid file name was specified. Make sure that TIBCO Rendezvous is available.
SXF7254E	<%s> Failed to append num of lines to genFilePublish message: cause=%s	Unable to add the number of lines field to a genFilePublish message. Make sure that TIBCO Rendezvous is available.
SXF7255E	<%s> Failed to append num of messages to genFilePublish message: cause=%s	Unable to add the number of messages field to a genFilePublish message. Make sure that TIBCO Rendezvous is available.
SXF7256E	<%s> Failed to create Tracking message of length=%d: %s	Unable to create a message. Contact the TIBCO administrator to ensure that TIBCO Rendezvous is functioning.

Number	Message	Description
SXF7257E	<%s> Failed to add filename to genFilePublish message: cause=%s	An attempt to add a field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF7258E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7259E	<%s> NULL string contents while retrieving value: %s, type: %d	The field type expected a STRING value but the received data contained a NULL instead. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7260E	<%s> Type mismatch while retrieving value: %s, type: %d	The specified field type does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7261E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7262E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7263E	<%s> Default Type mismatch while retrieving value: %s, from type=%d to type=%d	An attempt was made to retrieve a message, but the message type did not match the configuration file definition. Verify the type of field that was published.
SXF7264E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7265E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.

Number	Message	Description
SXF7266E	<%s> Type mismatch while retrieving value: %s, type: %d	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7267E	<%s> Unsupported FLOAT conversion of type=%d for %s	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7268E	<%s> Invalid data type flag <%04x> for <%s>	The DataType flag is invalid. Ensure that the field type is valid.
SXF7269E	<%s> Unsupported PACKED conversion of type=%d for %s	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7270E	<%s> String <%s> length is %d. Must be less than 19.	A string field was longer than 19 bytes. If a string field is to be used for numeric conversions, it must be less than 19 bytes.
SXF7271E	<%s> Unsupported ZONED conversion of type=%d for %s	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7272E	<%s> Unsupported COMP/SHORT conversion of type=%d for %s	The field type specified does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7273E	<%s> Unsupported COMP/INTEGER conversion of type=%d for %s	The specified field type does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7274E	<%s> Unsupported COMP/LONGLONG conversion of type=%d for %s	The field type specified does not match the received data. Correct the configuration file, and be certain that the configuration file matches the received message.

Number	Message	Description
SXF7275E	<%s> Unsupported FLOAT conversion of type=%d for %s	The field type specified does not match the received data. Correct the configuration file, and be certain that the configuration file matches the received message.
SXF7276E	<%s> Unsupported DISPLAY/INTEGER conversion of type=%d for %s	The specified field type does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7277E	<%s> Unsupported DISPLAY/UNSIGNED INTEGER conversion of type=%d for %s	The specified field type does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7278E	<%s> Unsupported DISPLAY/SHORT conversion of type=%d for %s	The specified field type does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7279E	<%s> Unsupported DISPLAY/UNSIGNED SHORT conversion of type=%d for %s	The specified field type does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7280E	<%s> Unsupported DISPLAY/INTEGER conversion of type=%d for %s	The specified field type does not match the received data. Correct the configuration file, and ensure that the configuration file matches the received message.
SXF7281E	<%s> Error saving current status for workfile, into statfile=%s	A file I/O error occurred. If the problem persists, contact your systems programmer.
SXF7282E	<%s> Could not malloc I/O buffer for move. cause=%d: %s	Unable to allocate memory. Report error to your TIBCO administrator.
SXF7283E	<%s> Move failed: could not open workfile: cause=%d: %s	The file specified could not be opened. Be sure that FilePublisher has the proper authority.
SXF7284E	<%s> Move failed: could not open target file <%s> cause=%d: %s	The specified file could not be opened. Make sure that FilePublisher has the proper authority.

Number	Message	Description
SXF7285E	<%s> Move failed: could not close workfile: cause=%d: %s	The specified file could not be moved. Make sure that FilePublisher has the proper authority.
SXF7286E	%s	The %s identifies the problem found.
SXF7287E	%s	The %s identifies the problem found.
SXF7288E	<%s> Move failed: Unable to close target file <%s> cause=%d: %s	The specified file could not be moved. Make sure that FilePublisher has the proper authority.
SXF7289E	<%s> Move failed: could not close workfile: cause=%d: %s	The specified file could not be moved. Make sure that FilePublisher has the proper authority.
SXF7290E	<%s> Move failed: could not open workfile. cause=%d: %s	The open operation failed. Check that the target directory is available and that there are no authority problems.
SXF7291E	<%s> Move failed could not open target file <%s> using mode=%s Cause=%d:	The open operation failed. Check that the target directory is available and that there are no authority problems.
SXF7292E	<%s> Move failed: Unable to close workfile: cause=%d: %s	The close operation failed. Check that the target directory is available and that there are no authority problems.
SXF7293E	<%s> Move failed: could not fwrite to target file<%s> Cause=%d: %s	The write operation failed. Check that the target directory is available and that there are no authority problems, and there is sufficient space on the volume.
SXF7294E	<%s> Move failed: could not fprintf to target file <%s> cause=%d: %s	The write operation failed. Check that the target directory is available and that there are no authority problems, and there is sufficient space on the volume.
SXF7295E	<%s> Move failed: could not write to target file <%s> Cause=%d: %s	The write operation failed. Check that the target directory is available and that there are no authority problems, and there is sufficient space on the volume.

Number	Message	Description
SXF7296E	<%s> Move failed: Unable to close target file <%s> cause=%d: %s	The close operation failed. Check that the target directory is available and that there are no authority problems.
SXF7297E	<%s> Move failed: Unable to close workfile: cause=%d: %s	The close operation failed. Check that the target directory is available and that there are no authority problems.
SXF7299E	<%s> Error moving working file to <%s>. Retry limit %d reached.	The target file was still locked out (in use by another job), after the max retry limit has been reached. The incoming file is converted to an ERR file, and trying to update the target file will be abandoned.
SXF7300E	<%s> Failed to retrieve sequence/block_id number from msg. cause=%s on Subject/Destination=%s	An attempt to retrieve a field from a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF7301E	<%s>: Failed to extract Hostname from end-of-file message	An attempt to retrieve a field from a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF7302E	<%s>: Failed to extract UserId from end-of-file message	An attempt to retrieve a field from an RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF7303E	<%s> Error saving STERR num error status file=%s	A file write failed. It is usually due to an out of space error. See the previous error message in the trace log for the exact cause.
SXF7307E	<%s> User=<%s> (Pub Id=%s) is not allowed WRITE access to target File=<%s>. Rejected. rc=%d	Security checking was requested, and the security facility indicated the incoming File Transfer request from the specified user, was not allowed access to publish that file. Contact your security people, to determine why the user is not authorized to write to that file.

Number	Message	Description
SXF7310E	<%s> Recovery for SEQ file failed probably because Ops deleted CWK work file=%s. Progress file is being reset and recovery for that file has been abandoned. cause=%d: %s	Unable to recover a work file being used to update a Sequential file. The file may be corrupted or may have been deleted. Contact the TIBCO administrator.
SXF7313W	FileType=<%s> Resetting PRG file %s to #NONE#	Denotes that a file transfer is complete, and that the file type is being reset.
SXF7315E	<%s> There is an error in the configuration for this FileType. Cannot create the subscriber %s	Unable to create a listener for a subscriber because of errors previously reported in the configuration processing phase. Check the trace log to determine the problem in the configuration file and correct it.
SXF7316E	<%s> UserId=<%s> from node=%s was not found in Security Mapping Table. Request to update Subscriber file was rejected. Block# %d	Security checking was requested, and the security facility indicated the incoming File Transfer request from the specified user, was not allowed access to publish that file. Contact your security people, to determine why the user is not authorized to write to that file.
SXF7317W	FileType=<%s> Deleting PRG file	Denotes that a file transfer is complete, and that the PRG file for that file type is being deleted.
SXF7318W	File Type <%s> Error encountered. Closing all .CWK, .PRG, and .PRP files	A severe I/O error occurred when processing a incoming file transfer. See the previous message in the trace log for the cause of the problem
SXF7319E	<%s> Rename of workfile %s to .ERR error file %s failed for %.16s. Reason=%d: %s	The rename operation failed. Check that the target directory is available and that there are no authority problems.
SXF7320E	<%s> Got I/O error on .PRG file=%s. Reason=%d: %s	A file write failed on a PRG progress file. It is usually due to an out of space error. See the previous error message in the trace log for the exact cause.

Number	Message	Description
SXF7328E	<%s> Failed to retrieve sequence/block_id number from Record ECM msg. cause=%s on Subject/Destination=<%s>. useNT=%d	An attempt to retrieve a field from a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF7329E	<%s> Rename of workfile %s to target file %s failed. Reason=%d: %s	The specified file could not be renamed. Be sure that FilePublisher has the proper authority.
SXF7331E	<%s> User=<%s> (Pub Id %s) is not allowed WRITE access to target File=<%s>. Rejected. RC=%d	A file is being written and an error was flagged (for example, the FileAdapter user does not have ALTER/Update authority to that file.). Contact your systems programmer.
SXF7332E	<%s> User=<%s> (Pub Id %s) is not allowed ALTER access to work file=<%s>. Rejected. RC=%d	A file is being written and an error was flagged (for example, the FileAdapter user does not have ALTER/Update authority to that file.). Contact your systems programmer.
SXF7334E	<%s> Unable to write to target=<%s> and 'File Locked' retry timer failed. cause=%d: %s	A file was locked and the File Adapter was unable to create a retry timer. This usually indicates an out of memory condition, or a failure/crash in the underlying RV support.
SXF7337E	<%s> File open failed because unable to allocate file. File is unavailable and/or allocated to another job.	The file is not available.
SXF7357E	Output File <%s> could not be opened using DD=<%s> mode=<%s> Type_alloc=%s errno=%d %s	The specified file could not be opened. The errno %s specifies the specific cause of the problem.
SXF7358E	Append Output File <%s> might not be currently cataloged on VOLSER=<%s ...>	The VOLSER parameter specified in the INI file may be incorrect for that file. Contact your systems programmer.
SXF7365E	Output File <%s> could not be opened using mode=<%s> errno=%d %s	A file open failed. The errno contains the cause of the error.

Number	Message	Description
SXF7367E	<%s> Failed to create nested EPM Payload message of length=%d: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF7368E	<%s> Unable to read source CWK file <%s> cause=%d: %s	The read operation failed. Check that the CWK workfile directory is available and that there are no authority problems.
SXF7369E	<%s> Unable to read target file <%s> cause=%d: %s	A file read failed. The errno contains the cause of the error.
SXF7370E	<%s> Subscriber I/O error caused File Transfer Abort: %d %s	A file write failed. It is usually due to an out of space error. The Abort code contains the cause of the error. The File Transfer is aborted.
SXF7371E	<%s> ConfBlock - Failed to create message of length=%d: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF7374E	<%s> File open, rename, or append write failed because unable to allocate file. File is unavailable and/or allocated to another job.	A file open or rename failed.
SXF7375W	<%s> Discarding Block # %d because of previous ECM I/O Error.	A new block was received after an I/O error was signalled to the Publisher. The block will be discarded.
SXF7376E	<%s> Failed to close .PRG file=%s, aborting renameByCopy. error=%d: %s	A file close failed for a PRG progress file. The errno contains the cause of the error. The file transfer will be aborted.
SXF7377E	<%s> Failed to open Progress file during Lock Retry: <%s> errno=%d errtext=%s	A file open failed during a locked file retry. The errno contains the cause of the error.
SXF7379E	<%s> There is an error in the configuration for this FileType. Will NOT perform the NT ECM Admin handshake for subscriber=%s ----	Check configuration phase error messages from FileAdapter startup and correct the error.

Number	Message	Description
SXF7392E	<%s> Error removing/deleting %s file=<%s>. Reason=%d: %s	Deletion of CWK file was unsuccessful.

SXF8000 – SXF8999

Number	Message	Description
SXF8001E	FileType=<%s>: Subject=<%s> Unable to extract reply_subject back to Publisher	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF8007E	FileType=<%s>: Subject=<%s> Publisher indicates we are not registered for ECM mode. Flipping out to non-ECM mode.	This normally indicates a configuration mis-match between the Publisher and the Subscriber.
SXF8008W	FileType=<%s>: Subject=<%s> Publisher rejected our VERIFY_IFACTIVE_REQUEST - we are not registered. Issuing SUB_ACTIVATION_REQUEST	This indicates either that the Progress file was deleted at the Publisher, or a configuration mis-match between Publisher and Subscriber.
SXF8010E	FileType=<%s>: Subject=<%s> Publisher rejected our SUB_ACTIVATION_REQUEST	This indicates either that the Progress file was deleted at the Publisher, or a configuration mis-match between Publisher and Subscriber.
SXF8015W	FileType=<%s>: Subject=<%s> Publishing ECM VERIFY_IFREGISTERED_ REQUEST handshake message with SubscriberName=%s using GUID=%s retry#%d	Message indicating that a restart/recovery sequence was performed between Publisher and Subscriber.
SXF8018E	<%s> Listener setup Failed for Admin Request subject <%s> cause=%s	An attempt to create a RV listener failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8019E	<%s> Listener setup Failed for Admin Reply subject <%s> cause=%s	An attempt to create a RV listener failed. The %s identifies the specific cause. It normally indicates insufficient memory.

Number	Message	Description
SXF8020E	NULL FileHandler in Admin timer callback	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF8021E	<%s>: Error on sending RV Admin message: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8022E	<%s>: Error on sending RV Admin message: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8023E	<%s>: Error on sending RV Admin message: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8024E	<%s>: Error on sending RV Admin message: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8025E	<%s>: Error on sending RV Admin message: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8026E	<%s>: Error on sending RV Admin message: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8027E	<%s>: Error on sending RV Admin message: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8028E	<%s> Admin Timer creation failed for Subject <%s>: cause=%s	An attempt to create a RV timer failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8029E	FileType=<%s>: Received unknown ECM Admin config reply. msgtype=%d	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF8030E	FileType=<%s>: SUB_IS_REG_REPLY = False. Startup Handshake rejected	This normally indicates a configuration mis-match between the Publisher and the Subscriber.

Number	Message	Description
SXF8031E	FileType=<%s>: Received unknown ECM Admin config reply. msgtype=%d	This is a Block Mode ECM internal or recovery type error. Contact TIBCO technical support.
SXF8032E	<%s> Failed to create RV Admin message: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8033E	<%s> Failed to append MsgType to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8034E	<%s> Failed to append MsgId/Guid to Admin message: %s	An attempt to add a new field to a RVmessage failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8035E	<%s> Failed to append Version to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8036E	<%s> Failed to append bool flag to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8037E	<%s> Failed to append NumRetries to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8038E	<%s> Failed to set RV Admin send subject: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8039E	<%s> Failed to set RV Admin reply subject: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8040E	<%s> Failed to send RV Admin msg: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.

Number	Message	Description
SXF8041E	<%s> Failed to create RV Admin SUB_ACTIVATION message: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8042E	<%s> Failed to append MsgType to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8043E	<%s> Failed to append MsgId/Guid to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8044E	<%s> Failed to append SubscriberName to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8045E	<%s> Failed to append Version to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8046E	<%s> Failed to set RV Admin send subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8047E	<%s> Failed to set RV reply subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8048E	<%s> Failed to send RV Admin msg: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8049E	<%s> Failed to create RV Admin REG_SUB_ALIVE_REPLY message: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.

Number	Message	Description
SXF8050E	<%s> Failed to append MsgType to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8051E	<%s> Failed to append MsgId/Guid to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8052E	<%s> Failed to append SubscriberName to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8053E	<%s> Failed to append Version to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8054E	<%s> Failed to set RV Admin send subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8055E	<%s> Failed to set RV reply subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8056E	<%s> Failed to send RV Admin msg: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8057E	<%s> Failed to create RV Admin VERIFY_IFACT message: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8058E	<%s> Failed to append MsgType to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.

Number	Message	Description
SXF8059E	<%s> Failed to append MsgId/Guid to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8060E	<%s> Failed to append SubscriberName to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8061E	<%s> Failed to append Version to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8062E	<%s> Failed to set RV Admin send subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8063E	<%s> Failed to set RV reply subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8064E	<%s> Failed to send RV Admin msg: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8065E	<%s> Failed to create RV Admin VERIFY_IFREG message: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8066E	<%s> Failed to append MsgType to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8067E	<%s> Failed to append MsgId/Guid to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.

Number	Message	Description
SXF8068E	<%s> Failed to append SubscriberName to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8069E	<%s> Failed to append Version to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8070E	<%s> Failed to set RV Admin send subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8071E	<%s> Failed to set RV reply subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8072E	<%s> Failed to send RV Admin msg: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8073E	<%s> Failed to create RV Admin VERIFY_IFACTIVE_REPLY message: %s	An attempt to create an RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8074E	<%s> Failed to append MsgType to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8075E	<%s> Failed to append Version to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8076E	<%s> Failed to append 'isactive' to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.

Number	Message	Description
SXF8077E	<%s> Failed to set RV Admin send subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8078E	<%s> Failed to send RV Admin msg: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.
SXF8079E	<%s> Failed to create RV Admin SUB_ACTIVATION_REPLY message: %s	An attempt to create a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8080E	<%s> Failed to append MsgType to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8081E	<%s> Failed to append Version to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8082E	<%s> Failed to append 'isactivated' to Admin message: %s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8083E	<%s> Failed to set RV Admin send subject: cause=%s subject=%s	An attempt to add a new field to a RV message failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF8084E	<%s> Failed to send RV Admin msg: cause=%s subject=%s	Unable to publish an ECM Admin message. Check that TIBCO Rendezvous is available.

SXF9000 – SXF9999

Number	Message	Description
SXF9002W	Library not found: %s	A library needed by the FileAdapter was not found.

Number	Message	Description
SXF9003E	API %s failed with exception %s	A TIBCO API failed. Note the return code and contact your TIBCO administrator.
SXF9004E	Input file <%s> does not exist	The input file specified does not exist. Correct the file name, or the library name.
SXF9005E	File <%s> is already present in process library	The specified file already exists in the process library. Move the specified file out of the process library to prevent the Adapter from overwriting the file.
SXF9006E	Library not found: %s	Could not find the specified library. Supply a name of a library that exists on the system.
SXF9007E	Input file <%s> does not exist	The input file specified does not exist. Correct the file name, or the library name.
SXF9008E	File <%s> is present in process library.	The specified file already exists in the process library. Move the specified file out of the process library to prevent the Adapter from overwriting the file.
SXF9502E	Generic error: %s %d %d %d %d	File Adapter is reporting an error it encountered. The %s details the cause of the error.
SXF9503E	Generic error: %s %s %d %d %d	File Adapter is reporting an error it encountered. The %s details the cause of the error.
SXF9504E	NULL SaveTimer in timer callback	An internal FileAdapter error occurred. Contact TIBCO technical support.
SXF9505E	SaveTimer Timer creation failed: cause=%s	An attempt to create a RV timer failed. The %s identifies the specific cause. It normally indicates insufficient memory.
SXF9506E	SaveTimer Timer removal failed: cause=%s	An attempt to free a RV timer failed. If the problem persists, then contact TIBCO technical support.
SXF9507E	Error renaming Sub log file <%s> to <%s> cause: %d %s. File logging turned off...	The File Adapter was unable to rename and open a new trace log file. The second %s details the cause of the problem.

Number	Message	Description
SXF9508E	Failed to open Sub log file <%s> cause: %d %s. File logging turned off...	The File Adapter was unable to open the trace log file. The second %s details the cause of the problem.
SXF9509E	Failed to open Sub log file <%s> cause: %d %s. File logging turned off...	The File Adapter was unable to open the trace log file. The second %s details the cause of the problem.

Index

Symbols

.LOG files 46, 80

A

adapter
  function overview 2
  relationship to IBM AS/400 2
  typical usage scenario 2
  usage scenarios 2
ADAPTER_NAME element 47, 81
AE wire format 3
AppendDateTime parameter 93
appendDateTime parameter 118
appendToExistingFile parameter 95
autoGenerateFile parameter 94, 118

B

binary datatypes 129
binary format 13
block transfer mode 118
blocksizeAlloc parameter 92
blockTransferMode parameter 58, 99, 118
blockTransferSize parameter 58, 118
BusinessEvents
  in introduction 14
  subject element for subscriber 81

C

certified messages delivery 130
checking trace logs 34
CL program importance 123
COBOL numeric data types 131
COBOL numeric datatypes 62
COBOL, emulating LOW-VALUES and HIGH-VALUES 63
codepage support 49
commands
  call 123
  DLT 36
  SBMJOB for FilePublisher 114
  SBMJOB for FileSubscriber 115
configuration file examples
  delimited file 66, 104
  different order formats 68, 106
  fixed-length file 67, 105
configuration files
  examples 40
configuration information, printing at startup 50, 83
configuring data sets 52
confirmationSubject parameter 59, 100
constant tag 63
constraint parameter 61, 61, 101
  tags 64
constructing subject name from data 129
containerName tag 64, 103
CONTINUE_ON_CONFIG_ERROR element 47, 81
convertTo tag 102
convertToString tag 62
customer support xi

D

daemon parameter 84
data blocks, limiting the number of 84
data sets
  configuration 52

dataSetType parameter 59
datasetType parameter 91
defaultTimeLimit parameter 85
DELETE_PUB_PRG_FILES element 47
DELETE_SUB_PRG_FILES element 81
delimited file
  configuration file examples 66, 104
  example 66
delimiter parameter 60, 99
delivery options 3
deliveryMode parameter 54, 54
description tag 101
different order formats 68, 106
double values
  publishing 130

E

ECM 10
  and RVCM 11
  error/restart handling 119
  FilePublisher 130
  RV subjects 120
ECM_NESTED_TEXT_LINES element 48
ECMSubscriberName parameter 59, 99, 121
EEM_DESTINATION element 48, 81
EEM_SUBJECT element 48, 81
EEM, support for 14
EMS_SESSION element 48, 51, 81, 82
endPublishDestination 57
endPublishSubject parameter 57, 118
EOL_CHARACTER element 52
EOL_ON_SUBSCRIPTION element 82
ERROR_DESTINATION element 48, 49, 82
ERROR_EXIT_CC element 48, 82
ERROR_SUBJECT element 48, 82
errors, writing to SYSLOG 89

examples
  configuration file for a delimited file 66
  configuration file for a fixed-length file 67
  configuration file for different order formats 68
  failed FilePublisher session 138
  failed FileSubscriber session 147
  FilePublisher 66
  FilePublisher configuration file for numeric data types 70
  FileSubscriber configuration file supporting numeric data types 107
  header record in an output file 132
  pre-processing 123
  SBMJOB for starting 115
  starting with sbmjob 114
  successful FilePublisher session 134
  successful FileSubscriber session 141
executeAfterProcess parameter 60, 98
executeBeforeProcess parameter 60, 98
exitOnFileSaveError parameter 95
explicit confirmation mode (ECM) 10

F

field parameter 101
fieldStart tag 63, 102
file transfer
  ECM 14
FILE_COUNT element 43, 75
FILE_LIMIT element 43, 75
FILE_LINE element 101
FILE_NAME element 43, 75
FILE_OPTIONS element 53, 90, 131
fileHeader parameter 98, 132
  example 132
filePrefix parameter 53, 90

FilePublisher
configuration file 43
ECM 130
example 70
example of failed session 138
example of successful session 134
examples 66
numeric data types 70
overview 5, 5
polling method 6
post-processing of files 124
pre-processing of files 123
sample configuration files 41
starting 114, 114
stopping 116
usage guidelines 66
FileSubscriber 8
configuration file 75
example for COBOL numeric data types 107
example of failed session 147
example of successful session 141
examples 104
numeric data types exampe 107
post-processing of files 124
pre-processing of files 123
sample configuration files 41
starting 115, 115
stopping 117
usage guidelines 131
fileTrailer parameter 98, 132
FileType section 53, 90
fixed-length file
configuration file examples 67, 105
example 67
floating-point datatypes 129
forcePublishedFileName parameter 96
fromMessage tag 102
functional components 4

G

GDG
output files, storing 88

generateFileDestinationName parameter 97
generateFileFieldName parameter 57, 97
generateFileOnNumberOfMessages parameter 94
generateFileOnNumMsgs parameter 118
generateFileSubjectName parameter 97, 118
genFilePublishDestinationName parameter 96
genFilePublishSubject parameter 96
grouping
fields 64
progress (.PRG) files 83
work (.CWK) files 89

H

header record in an output file 131
heartbeat messages 125
format 125
HEARTBEAT_FILE_INFO element 49, 82
HEARTBEAT_TIME element 49, 82, 125
HIGH-VALUES (COBOL) 63
HOST_CODEPAGE element 49, 83

I

info messages 154
INPUT_DATASET element 49
inputDataset parameter 59
installation
requirements 22
isBinary parameter 61, 99
isCertified parameter 60, 97

J

JCL consideration 75
JMS_TIBCO_MSG_TRACE parameter 54, 54, 90

K

keepTrailingBlanks parameter 60
KILL_PUBLISHER element 49

L

label tag 62
ledgerFile parameter 85
length tag 63, 64, 102
lineLength parameter 59, 92
lineLength tag 103
listenerName parameter 52
listeningSubject
  parameter 52
location tag 63, 65
logMatched parameter 61
logs
  trace 34
logUnmatched parameter 61
LOW-VALUES (COBOL) 63

M

mapping user ids 86
MAX_CONCURRENT_JOBS element 49
memory
  preventing exhaustion 84
MESSAGE_FIELDS element 61
messageContainer parameter 61
  tags 64
messageItem parameter 61
messagesPerTransaction parameter 56
mode processing, recording 86
multiple record formats 64
MVS volume 46, 80, 88, 88, 88, 89

N

name parameter 84
name tag 64
nested message in a container 64
network parameter 84
NETWORK_CODEPAGE element 50, 83
noOfRetries parameter 94
noWaitAfterConfirmations 120
noWaitAfterConfirmations parameter 59
numeric data types 131

O

opaque data 129
opaque data types 129
operating system supported 22
Options section 47, 80
order formats 68
  configuration file example 106
  configuration file examples 68
  example 68
output file
  creation 9
  format 9
  header and trailer 131
output file creation 94
OUTPUT_DATASET element 50, 83
outputDataset parameter 59, 92

P

packed datatypes 129
packed format 13
  formula for 13
padCharacter parameter 98
padCharacter tag 102
padDirection parameter 98
padDirection tag 102

parameters
  alwmltthd 115, 115, 116
  executeAfterProcess 123
  executeBeforeProcess 123
  iscertified 97
  jobmsgqfl 114, 115, 116
PERSIST_EMS_ECM_MSGS element 50
polling method 6
pollInterval parameter 56
position tag 63, 64, 102
precision 13
precision tag 62, 102
Pre-Register section 52
pre-registering subscribers 130
primaryAlloc parameter 92
PRINT_FILE_OPTIONS element 50, 83
PRINT_STDOUT element 43, 75
PROCESS_DATASET element 50
processDataset parameter 59
progress (.PRG) files
  storing 89
progress (.PRG) files, grouping 83
PROGRESS_DATASET element 83
PUBLISH_HEARTBEAT element 50, 83, 125
publishDestinationName parameter 53, 54
publishDestinationType parameter 53, 54
publishing double values 130
publishSubjectName parameter 53

Q

QUEUE_LIMIT element, subscriber 84
queueing of files 5

R

receiving COBOL numeric data types 131
removeAfterProcess parameter 60
removeLeadingBlanks parameter 60
removeTrailingBlanks parameter 60
requireOldMessage parameter 52
requireOldMessages parameter 85, 130
retransmissionDelayTicks 120
retransmissionDelayTicks parameter 58, 100
retryInterval parameter 95
RV vs RVCM 3
RV_SESSION element 50, 84
RVCM
  vs. ECM 11
RVCM_SESSION element 51, 85
rvcmTimeLimit parameter 60

S

sample configuration files 40
sample data files 42, 42
sample files for configuration 40
saveFileInterval parameter 94
secondaryAlloc parameter 92
SECURITY_CHECK_FILE element 86
sending COBOL numeric data types 128
sending data untranslated (opaque) 129
SEQ (sequential)
  files, storing 88
service parameter 84
session type 47, 47
skipPadding parameter 98
software requirements 22
startAtLine parameter 61
starting FilePublisher 114
starting FileSubscriber 115
startNewMessage tag 64
startPublishDestination parameter 56
startPublishSubject parameter 56
stopping FilePublisher 116
stopping FileSubscriber 117
subject name
  constructing 129
subscribeDestinatinType parameter 90
subscribeDestinationName parameter 90
subscriber
  pre-registering 130
SUBSCRIBER element 52
subscribeSubjectName parameter 90

support, contacting xi
supported data types 38
supported file types 38
syncLedger parameter 85
SYSLOG, writing erros to 89

T

tags
  in the constraint parameter 64
  in the messageContainer parameter 64
  in the messageitem parameter 62
  precision 13
technical support xi
TERMINATE_ON_RV_SEND_FAIL element 51, 86
testing the adapter 28
TIBCO BusinessEvents 14
TIBCO Rendezvous 4
  subjects used by ECM 120
tibrvsend API, cannot call 86
trace files 46, 80
trace logs 34
Trace section 43, 75
TRACE_EMS_EPM_ERROR_MSGS element 44, 76, 77, 78
TRACE_EMS_HEARTBEAT_MSGS element 44, 76, 77, 78
TRACE_HEAP element 44, 76, 77, 78
TRACE_LEVEL element 45, 79
TRACE_SWITCHES element 46, 80
trackingIdDestination Name parameter 55, 91
trackingIdSubject parameter 55, 91
trailer record
  in output file 131
transactionDelay parameter 56
transferType parameter 58
translation table, specifying 49
transport mechanisms 3
triggerDestinationName parameter 56
triggerDestinationType parameter 56
triggerFieldName parameter 56
trigger-subject method 7
triggerSubjectName parameter 56
truncateRecords parameter 92
type tag 62, 102

U

UNIT_CWK element 87
UNIT_GDG element 87
UNIT_OUTPUT element 87
UNIT_PRG element, subscriber 87
UNIT_TRACE element, publisher 46
UNIT_TRACE element, subscriber 80
usage guidelines
  FilePublisher 66
  for publisher and subscriber 123
useExplicitConfirmation 120
useExplicitConfirmation parameter 58, 99, 120
useFieldWidth parameter 60
useFilePolling parameter 56
useFixedRecordFile parameter 92
userids, mapping 86
useTrackingId parameter 55, 91

V

value tag 63, 64, 102
value tag for MESSAGE_FIELDS parameter 63
VOLSER_CWK element 88
VOLSER_GDG element 88
VOLSER_OUTPUT element 88
VOLSER_PRG element, subscriber 89
VOLSER_TRACE element, publisher 46
VOLSER_TRACE element, subscriber 80
volume, MVS 46, 80
vsamFileMode parameter 100
vsamLogFile parameter 100
vsamUseLog parameter 101

W

wire format 3
work (.CWK) files
- grouping 89
- storing 88

WORKFILE_DATASET element 89
WRITE_TO_SYSLOG element 51, 89

Z

zoned datatypes 129
zoned format 13