



TIBCO FTL® - Enterprise Edition

Glossary

Version 7.0.1 | December 2024

Contents

Contents	2
About this Product	3
Glossary	4
TIBCO Documentation and Support Services	19
Legal and Third-Party Notices	21

About this Product

TIBCO® is proud to announce the latest release of TIBCO FTL® software.

This release is the latest in a long history of TIBCO products that use the power of Information Bus® technology to enable truly event-driven IT environments. TIBCO FTL software is part of TIBCO Messaging®. To find out more about TIBCO Messaging software and other TIBCO products, please visit us at www.tibco.com.

Glossary

A

Ability

An endpoint can carry messages using up to four abilities: 1. One-to-many receive (subscriber), 2. One-to-one receive (inbox subscriber), 3. One-to-many send (send call), 4. One-to-one send (send-to-inbox call). Administrators bind transports to specific endpoint abilities. [Development, Administration]

ACL

An access control list which is a list of permissions associated with a system resource. An ACL specifies the users or system processes that are granted access to objects, as well as what operations are allowed on given object.

Advisory

TIBCO FTL software uses asynchronous advisory messages to convey errors, warnings or other information. [Development]

Application

1. An FTL application is an executable program that communicates using TIBCO FTL software. 2. Within the realm definition, an application (or more explicitly, an application definition) configures the endpoints, formats, transports, and stores for an application program. [Concepts, Administration]

Application Definition

An application definition is a configuration within the realm that represents an application program, declares its endpoints and formats, and configures endpoint implementation for its process instances. An application definition consists of a name, a set of endpoints, a set of application instance definitions, and a set of preload formats. [Concepts, Administration]

Application Instance Definition

An application instance definition configures endpoints for use by application processes in a specific context, for example, a process that runs on a specific host computer or supplies an application instance identifier. Structurally, an application

instance definition consists of a name, a set of match parameters, and a set of connectors to implement each endpoint. [Concepts, Administration]

Application Instance Identifier

Administrators arrange transports to implement endpoints in different ways, and select among them at runtime using an application instance identifier. Application programs obtain that identifier from the command line, a configuration file, or an environment variable, and then supply that identifier as a property value in the realm connect call. [Concepts, Development]

Apportioning Message Streams

A store can apportion a message stream among a set of cooperating subscribers. Only one subscriber receives and processes each message. Acting together, the subscribers process every message in the stream. [Concepts]

Authentication Service

A service used to authenticate the identity of, then authorize, client programs, administrative tools, and affiliated servers, typically as a prerequisite to allowing access or communication.

Auxiliary Servers

Optional auxiliary servers are configured for a specialized purpose. It is not designated in the list of core.servers. An auxiliary server might provide an eFTL service to expand capacity, backup bridge service for fault tolerance, or provide a persistence service in a disaster recovery model. Also see Primary Servers and Satellite Servers. [Administration]

B

Bridge

See Transport Bridge.

Broker

A message broker is an FTL server that provides server-based messaging using a message broker pattern. [Concepts, Administration]

Broker Based Messaging

In broker based messaging, publishers send messages to the FTL server, and messages are held in an intermediate store until they are successfully sent to

subscribers. This model allows for adjustment of reliability and throughput parameters, and provides flexible administration and monitoring. [Concepts]

Brokerless Messaging

In brokerless messaging, publishers send messages directly to subscribers, with generally lower latency than when using a broker for storage. The FTL server still provides some management for the client applications. [Concepts]

C

Certificate SAN (Subject Alternate Name)

A Subject Alternate Name (or SAN) certificate is a digital security certificate which allows multiple hostnames to be protected by a single certificate. Also called a Unified Communication Certificate (or UCC), a multi-domain certificate, or an Exchange certificate.

Cluster

A cluster of FTL Servers which can be arranged to provide fault tolerant services, including persistence services, realm services, group services, and bridge services. See Persistence Clusters. [Concepts]

Content Matcher

A content matcher is a pattern matching template for message fields and values. A subscriber can use a content matcher to narrow its interest to a subset of a message stream. [Concepts, Development]

Core Server

A core server is an FTL Server that you designate for managing and storing realm deployments. Also see Primary Servers, Satellite Servers, and Auxiliary Servers. [Administration]

D

Delivery Assurance

A store can retain message streams for a disconnected subscriber, assuring delivery after the subscriber returns. The subscriber receives the complete message stream. [Concepts]

Deploy

In FTL, an administrator modifies the realm definition so the FTL server's clients can use the new realm definition. The updated definition put into production use is said to be deployed. [Administration]

Direct Path

The direct path is a communications path in which messages travel directly between publishing and subscribing endpoints, without an intermediate stop at a store. In contrast, an indirect path goes through a store, which may optionally back a direct path. [Concepts, Administration]

Disable

An administrator can use the FTL server GUI to disable a client process. Disabled clients cannot use TIBCO FTL communications resources. [Administration]

Disk-Based Persistence

Enable disk-based persistence to store FTL messages and metadata to disk, when disk is more cost effective than memory or when a stronger degree of persistence is desired. The stored messages and metadata can be automatically recovered on a full restart of the persistence cluster. [Concepts]

Dispatch

Programs use API calls to dispatch inbound messages from event queues to callback methods. [Development]

Durable

A durable is the data structure within a store that maintains subscriber interest, forwards messages, and tracks delivery acknowledgments. There are different types of durables. See Standard Durable (with Prefetch Enabled), Standard Durable (no Prefetch), Shared Durable, Last-Value Durable, Dynamic Durable, Static Durable, and Map Durable. [Concepts , Administration, and Development]

Durable Subscriber

A durable subscriber is a subscriber object in an application process that maintains its identity even if the process exits and restarts. The subscriber relies on its durable to ensure message delivery. [Concepts, Development, Administration]

Dynamic Durable

A dynamic durable uses a dynamic durable template to enable durable subscribers within programs to create a set of durables at run time. [Concepts , Administration, and Development]

Dynamic Durable Template

An administrative configuration that supplies parameter values for dynamic durables. [Administration]

Dynamic Format

When a program constructs a self-describing message, its format is dynamic. That is, the format changes each time the program adds a field to the message. The message's metadata reflects the current state of the dynamic format. Also see Managed Format. [Development]

E

eFTL

TIBCO eFTL software is an add-on product that extends TIBCO FTL messaging to platforms that cannot support the full TIBCO FTL bus. Platforms such as web browsers and mobile phones can use TIBCO eFTL software to communicate with back-end infrastructure based on TIBCO FTL software.

eFTL Service

An eFTL service is an FTL service used to integrate mobile and browser platforms into an enterprise messaging solution.

Endpoint

An endpoint is the connection point between a transport and the publisher, subscriber, or broker. An endpoint contains transport details and insulates programs, and their designers/users, from those details. This enables separation of responsibilities between developers and administrators. Many publishers and many program processes may publish on one publishing endpoint. One subscriber may consume from more than one publishing endpoint. A program can have more than one endpoint to tap into more than one transport message stream. [Concepts]

Event Queue

Event queues buffer inbound messages during message delivery. Programs dispatch messages from event queues to callback methods. [Development]

F

Fault Tolerance

Fault tolerance is a system's ability to continue operating uninterrupted when one or more of the system's components fail. [Concepts]

Field

Each field of the metadata for a message has a field name and a field type. Each field of the message contains the data value. [Concepts, Development]

Format

A format defines the set of fields that a message can contain, including field names and data types. The format is available to cooperating programs before they exchange messages. Messages only carry data, which reduces both the message size, transport bandwidth, and processing time. [Concepts, Development, Administration]

Forwarding Zone

Forwarding zones delineate the scope of wide-area durables. A forwarding zone contains a set of persistence clusters, which cooperate to forward wide-area durables among all the persistence clusters in that zone. [Concepts, Administration]

FTL Server

An FTL server is the main point of contact for FTL clients, providing clients with services. Its configuration file determines the services that it provides. [Development, Administration]

G

Group

A group is a named set of application processes. A group service tracks group members as the processes join and leave the group and connect and disconnect from the FTL server. The group membership functionality is used by cooperating applications that belong to the same group to coordinate fault-tolerant operations or to designate operating roles among the cooperating application processes. [Concepts, Development]

Group Service

FTL's group service coordinates fault-tolerant operation and operating roles among applications processes. Every core FTL server automatically provides a group service. Application processes in a group use the group facility's API calls to interact with the group service, and to coordinate appropriate behavior. Group member applications and the group service are all within the same realm, and usually run on host computers within a local area network. Do not use the group facility across routers and WANs. [Concepts, Development]

I

Immutable

A message is immutable if a program cannot modify the fields of the message. [Development]

Inbound

Inbound indicates the direction proceeding into an endpoint from a transport, or into a program through an endpoint. For example, subscriber objects receive inbound messages. [Development, Administration]

Inbox

An inbox is a data structure that identifies a unique inbox subscriber object, enabling one-to-one communication to that subscriber. [Concepts, Development]

Inbox Subscriber

An inbox subscriber can receive one-to-one communication. An inbox subscriber is one kind endpoint instance within a program. An inbox subscriber instantiates the one-to-one receiving capabilities of an endpoint. (Contrast Subscriber.) The term "subscriber" is used in a one-to-many model whereas an "inbox subscriber" is used in one-to-one communication. [Concepts, Development]

Indirect Path

An indirect path is a communications path in which messages travels with an intermediate stop at a persistence store. The indirect path optionally backs a direct path. [Concepts, Administration]

Inline Mode

Programs that receive time-sensitive messages can use inline mode to favor low latency over high throughput. Restrictions apply. [Development, Administration]

Interest

A subscriber expresses interest in the message stream arriving at an endpoint to potentially receive the message. If the subscriber has a content matcher, it expresses interest in a subset of that message stream: namely, the subset with matching content. A durable expresses interest on behalf of its durable subscriber, even when the subscriber is absent. [Concepts, Development, Administration]

Iteration

Iteration is a sequence of instructions are repeated a specific number of times or until a condition is met. When the first set of instructions is executed again, it is called an iteration.

K

Key-Value Map

A key/value map stores key/value pairs where each key is a string and a key's value is a message. Programs can use map methods to store and retrieve key/value pairs, and to iterate over the pairs in a map. This is a way to use the store as a simple database table. [Concepts]

L

Last-Value Durable

A last-value durable is a durable that represents the interest of potentially many subscriber objects. It stores only the most recent message from one or more message streams. Use a last-value durable to hold recent context for initializing new subscribers.

Latency

Latency is the time for a message to be transferred between its original source and its destination. Latency is typically measured in milliseconds. [Concepts]

M

Managed Format

A message format is a fixed set of field names and field types. This metadata describes message content and is provided to cooperating programs before they exchange messages. A message that conforms to a format carries only data values, which reduces both the message size and processing time. A message format is

defined in the realm definition. Also see Dynamic Format. [Concepts, Administration]

Many-to-Many

Many-to-many communication is when many publishers send messages to many subscribers and, potentially, every subscriber can receive messages from each of the publishers.

Many-to-Many or One

Many-to-many or one communication is when two or more publishers publish messages and one or more subscribers receive messages. [Concepts]

Map API

A MAP API is a set of methods for storing and retrieving key/value pairs in a last-value durable. [Development]

Map Durable

A map durable is a durable that stores a key/value mapping. When a message arrives, a map durable uses the value in the message's key field as the key, and stores the full message as its value. For each distinct key, the durable stores only the most recent value. (One map durable can hold many key/value pairs.) [Concepts, Administration, and Development]

Message

In FTL, a message format defines the metadata which is the set of fields that a message can contain. The message only carries data. This approach reduces the message size, transport bandwidth, and processing time. For efficiency, TIBCO FTL messages support a limited set of data types. Publishers send messages and subscribers receive messages through endpoints. A program can send messages, receive messages, or both. [Concepts]

Message Stream

A message stream is a sequence of messages. A publisher is the source of the message stream, and a subscriber expresses interest in that stream. [Concepts]

Message Swapping

Messages held in process memory can be swapped to disk when configured memory limits are exceeded. Excess messages are temporarily written to disk, as needed. [Concepts and Administration]

Multi-tenancy

Multiple independent instances of one or multiple applications operate in a shared environment.

Mutable

A program may modify the fields of a mutable message. [Development]

N

Notification

A notification is an out-of-band communication from TIBCO FTL software to an application program. TIBCO FTL software delivers notifications when conditions preclude delivering advisories using event queues. [Development]

O

One-to-Many Publishing

One-to-many publishing is a common model in TIBCO FTL where a publisher produces a stream of messages and the subscribers can receive messages, sometimes selected by content matching. [Concepts]

One-to-One Communication

One-to-one communication lets a publisher send a message to a specific inbox subscriber which is a kind of endpoint instance. Only that inbox subscriber receives the message, even when other subscribers share the transport. [Concepts]

Outbound

Outbound indicates the direction proceeding out from an endpoint to a transport, or out from a program through an endpoint. For example, publisher objects send outbound messages. [Development, Administration]

P

Permissions

Permissions are set at the cluster and store level to grant access to objects for given user/role.

Persistence

Persistence is the potential to store a message after sending it and before it is received so off line subscribers can consume it at a later time. Persistence services use stores and durables to implement this potential. See also Apportioning Message Streams and Delivery Assurance. [Concepts, Administration, Development]

Persistence Clusters

The persistent services of a cluster of FTL servers cooperate in a persistence cluster. A persistence cluster maintains a set of persistence stores, replicating data for fault tolerance. [Concepts]

Persistence Service

A persistence service is an FTL service used to manage stores and durable subscriptions (durables).

Preload Format

Preload formats are the managed formats available to an application program. [Administration]

Primary Servers

Primary servers form the central hub within a family of affiliated servers and operate within a local network. Servers in the primary cluster function as interchangeable peers. Also see Satellite Servers and Auxiliary Servers. [Administration]

Primary Set

To prepare for disaster recovery, a persistence cluster can define two sets of persistence services, with some services at the main site and others at the disaster recovery site. The primary set is the set of active persistence services at the main site, while the standby set is the set of persistence services at the disaster recovery site.

Privileged Users

Privileged users are users who have been delegated extra levels of control. Users who lack most privileges are defined as unprivileged.

Publisher

A publisher sends messages and is the source of a message stream. A publisher instantiates (verifies) the send capabilities of an endpoint. Within a program, a

publisher is one kind of endpoint instance. [Concept]

R

Realm

A realm represents one communication fabric, with its endpoints, applications, application instances, transports, stores, and message formats. [Concepts, Development, Administration]

Realm Service

A realm service is an FTL service used as a unified communication fabric within and across networks which supplies application programs with communication configuration data. A realm service maintains a realm definition, which the FTL server distributes to client applications. [Development, Administration]

Request/Reply

Request/reply is a communication pattern where a requester sends a request message to a replier system, which receives and processes the request, ultimately returning a message in response. Request/reply can be one-to-many or one-to-one communication. [Concepts]

S

Satellite Servers

Optional satellite servers provide local service to clients that are geographically distant from the primary servers, or clients that are insulated from the primary servers (to fulfill enterprise requirements). Also see Primary Servers and Auxiliary Servers. [Administration]

Self-Describing Message

A self-describing message does not conform to a predefined format. A publishing program assembles a message with any number of fields and any combination of the data types. Each message contains the name and data types of each field which enlarges each message and results in slower processing of messages. Contrast to FTL's formatted messages (see Format). [Concepts]

Send

Programs make send calls to send messages. More narrowly, the term send also refers specifically to one-to-many publishing. (Contrast Send-to-Inbox.) [Concepts]

Send-to-Inbox

Programs make send-to-inbox calls to send one-to-one messages. (Contrast Send.)
[Concepts]

Shared Durable

A shared durable is a durable that represents the identical interest of two or more cooperating subscriber objects, apportioning a message stream among those subscribers. Use a shared durable when a set of subscribers collectively process a message stream in a distributed or parallel fashion, processing every message exactly once. [Concepts , Administration, and Development]

Standard Durable (no Prefetch)

A standard durable represents the interest of one subscriber object and forwards messages to the subscriber on request. With prefetch disabled (no prefetch), messages typically flow directly from publisher to subscriber and the store serves as a backup. [Concepts , Administration, and Development]

Standard Durable (with Prefetch Enabled)

A standard durable represents the interest of one subscriber object and forwards messages to the subscriber on request. With prefetch enabled, the store functions as a message broker, with no messages traveling peer-to-peer from publisher to subscriber. [Concepts , Administration, and Development]

Static Durable

A static durable is defined in advance of any subscribers and is defined in the realm configuration. [Concepts , Administration, and Development]

Store

A store holds messages until they are consumed. A persistent store is a store that is optionally replicated to further ensure against loss of messages. Administrators can configure stores within a persistence cluster or forwarding zone. Each store collects and maintains exactly one message stream and can collect the stream from one or more publishers. The store can maintain that message stream for one or many subscribers. [Concepts, Administration]

Store Inboxes

Store inboxes use the application's endpoint configured store or system store to store inbox data. When the endpoint's configured store is used, this is called an

endpoint store inbox. Otherwise, the dedicated inbox stores `ftl.system.inbox.store` or `ftl.routing.inbox.store` are used. [Concepts, Administration]

Subscriber

A subscriber expresses interest in a message stream and receives messages. A subscriber instantiates (verifies) the receiving capabilities of an endpoint. Within a program, a subscriber is one kind of endpoint instance. Also see "Inbox Subscriber". [Concepts]

T

Template

See Dynamic Durable Template.

Throughput

Throughput is the number of messages processed in a given time over a communication channel. Often measured messages per second. [Concepts]

Transport

Messages travel from publishers to subscribers over a transport which is a shared communication medium. TIBCO FTL supports a variety of transport types to use based on messaging requirements and environment. Each transport is a separate communication medium, insulated against crosstalk from other transports. [Concepts, Administration]

Transport Bridge

A transport bridge is a specialized process that can forward messages among sets of transports. [Administration]

Transport Bridge Service

A transport bridge service is an FTL service used to forward messages among multiple transport types.

TTL (time-to-live)

A TTL (time-to-live) defines how long a message remains in a system. Sometimes used to determine how long redelivery should be attempted if a subscriber cannot initially be reached.

U

Unprivileged User

Users who lack most privileges are defined as unprivileged. Users who have been delegated extra levels of control are called privileged.

V

Variant

See Application Instance. [Concepts]

W

Wide-Area Durable

Wide-area durables are accessible to publishers and subscribers across a wide-area network (WAN) link as if the durable is in a local persistence store. [Concepts, Administration]

Z

Zone

See Forwarding Zone. [Concepts, Administration]

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join [TIBCO Community](#).

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

Documentation for TIBCO FTL® - Enterprise Edition is available on the [TIBCO FTL® - Enterprise Edition Product Documentation](#) page.

TIBCO eFTL™ Documentation Set

TIBCO eFTL software is documented separately. Administrators use the FTL server GUI to configure and monitor the eFTL service. For information about these GUI pages, see the documentation set for TIBCO eFTL software.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

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

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document 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 Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, FTL, eFTL, and Rendezvous are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or 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. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

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

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. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

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 "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 2009-2024. Cloud Software Group, Inc. All Rights Reserved.