TIBCO FTL® - Enterprise Edition

Release Notes

Version 7.0.1 | December 2024

Contents

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.

New Features

The following features have been added in this recent releases of TIBCO FTL® - Enterprise Edition.

7.0.1

OpenSSL 3.0.15 Support

TIBCO FTL® - Enterprise Edition now supports OpenSSL 3.0.15

Disaster Recovery for Routes

The disaster recovery feature may now be used with routed persistence clusters. For details, see Disaster Recovery for Routes in FTL Administration.

In addition, the "suspend" REST API command can be used with the disaster recovery feature for planned failback, regardless of whether routing is configured. For details, see POST cluster in FTL Administration.

I/O metrics for disk persistence

Added metrics to track low-level I/O activity for persistence services that are configured for disk persistence. The metrics are:

```
TIB_MONITORING_TYPE_DISK_WRITE_COUNT
```

```
TIB_MONITORING_TYPE_DISK_WRITE_BYTES
```

```
TIB_MONITORING_TYPE_DISK_READ_COUNT
```

```
TIB_MONITORING_TYPE_DISK_READ_BYTES
```

```
TIB_MONITORING_TYPE_DISK_FDATASYNC_COUNT
```

All metrics are available via the monitoring stream or prometheus endpoint.

Latency stats for persistence

Added metrics to measure round-trip latency from the persistence cluster leader to each replica, as well as the latency of disk writes in each persistence service.

For more details, see FTL Prometheus Metric Naming and Catalog of Metrics in FTL Administration.

Reduced IOPS Utilization

FTL disk persistence now generally uses fewer IOPS (e.g. in cloud environments). EMS using FTL stores also sees improvement in IOPS usage. The improvements are focused on medium-sized messages (a kilobyte or so).

Persistence service can now trace messages from select clients

To enable persistence message tracing, set the loglevel of the persistence services to "msg:debug", and set the loglevel of the client to "msg:debug". The persistence service only traces messages from clients with "msg:debug" set; messages from other clients are not traced.

For more details, see Message Tracing

7.0.0

Added a new monitoring metric, "no_match_msg_count"

Added a new monitoring metric, `"no_match_msg_count"`. This metric tracks the number of messages sent to a persistence store that did not match any durables. Messages that do not match are dropped by the persistence service. In C API, the metric is `TIB_MONITORING_TYPE_NO_MATCH_MSG_COUNT`.

Added cumulative connection counts for FTL and eftl

Added a monitoring metric to track the total number of successful FTL client connections to an FTL server. In C API, the new metric is `TIB_MONITORING_TYPE_FTL_CUMULATIVE_CONNECTION_COUNT`.

OpenSSL 3.0.13 Support

TIBCO FTL® - Enterprise Edition now supports OpenSSL 3.0.13.

Retention and Replay of messages from a durable

You can configure a retention time on shared durable and standard durables with prefetch. When retention is enabled, acknowledged messages are retained until the retention time has elapsed.

When retention is enabled, users may rewind a durable using the user interface, REST API or the client API. Rewinding a durable allows new consumers to see a replay of messages that are published within the retention time.

For more information, see "Retention Time" in *TIBCO FTL®Administration* guide.

No Local Delivery

You are allowed to create no-local durable subscribers.

To create a no-local durable subscriber, pass the `TIB_SUBSCRIBER_PROPERTY_BOOL_NOLOCAL_MESSAGE_DELIVERY` property or API equivalent to the subscriber create call.

A no-local durable subscriber works like an ordinary subscriber, except that it does not receive messages published by the same TIBCO FTL client, even if the message matches the subscriber's interest.

A no-local durable subscriber may only be created on standard durables (with or without prefetch).

For more information, see "No-Local Message Delivery" in *TIBCO FTL®Development* guide.

Improved User Interface

The administrative GUI now has new options on the following pages:

- Realm Properties Details panel
- Durable Details panel

For more information, see "Durable Details Panel" and Realm Properties Details Panel in *TIBCO FTL®Development* guide.

Authentication mode

User-Defined Certificates with TLS

TIBCO FTL now allows the user to provide certificates for use with TLS connections to TIBCO FTL server, rather than relying on TLS certificates generated by TIBCO FTL.

> **Note:** In this mode, secure peer-to-peer transports (transports for direct communication between application clients) are not permitted.

Migrating from FTL-generated certificates to user-defined certificates requires a special procedure.

For details, see "Eliminating FTL-Generated Certificates" in *TIBCO FTL® Administration*.

Permissions without TLS

TLS is no longer required when enabling permissions for TIBCO FTL persistence. Users can enable authentication and permissions in TIBCO FTL (but not TLS), and then secure the network through other means.

For details, see "Ensuring FTL System Security: Tasks for Administrators" in *TIBCO FTL® Administration*.

TLS Termination for Client Connections

Instead of enabling TLS at TIBCO FTL server, users can provide a TLS certificate to an ingress point that terminates TLS. Clients should be configured to use TLS as normal.

Authentication and permissions are supported in this configuration.

This configuration is not supported for connections between TIBCO FTL servers. TLS termination is only supported for connections from clients and administrative tools.

For details, see "Ensuring FTL System Security: Tasks for Administrators" in *TIBCO FTL® Administration*.

Built-In LDAP Authentication

Added a new built-in authentication provider that allows TIBCO FTL server to authenticate incoming connections with an LDAP server.

For details, see "Using the Built in LDAP Authentication Service" in *TIBCO FTL® Administration*.

Built-In mTLS Authentication

When TLS is enabled with user-defined certificates, clients may authenticate to TIBCO FTL server with a TLS certificate. The common name (CN) of the certificate must be in a specific format.

For details, see "Using the Built in mTLS Based Authentication Service" in *TIBCO FTL® Administration*.

> **Note:** mTLS authentication is not supported for the UI.

Built-In OAuth 2.0 Authentication

Clients may authenticate to TIBCO FTL server using a signed JWT token issued by an oauth server.

For details, see "Using the oAuth 2.0 Authentication Service" in *TIBCO FTL® Administration*.

If upgrading from 6.x, and oauth 2.0 authentication with TLS is desired, users should consider switching from FTL-generated certificates to user-defined certificates. This allows FTL to enforce token expirations.

For details, see "Enabling TLS for FTL Server" in *TIBCO FTL® Administration*.

OAuth 2.0 based SSO for TIBCO FTL server UI.

TIBCO FTL support SSO for TIBCO FTL server UI, when the TIBCO FTL server is configured with OAuth 2.0 Authentication.

Multiple Authentication Providers

It is now possible to configure TIBCO FTL server to use multiple authentication providers.

For details, see "Authentication and Authorization" in *TIBCO FTL® Administration*.

Mapping Authorization Groups

If the configured authentication provider cannot return the desired authorization groups, TIBCO FTL server can map the provider's authorization groups to different ones for use in TIBCO FTL.

For details, see "Mapping Authorization Groups" in *TIBCO FTL® Administration*.

Prometheus Endpoints

TIBCO FTL server supports integration with Prometheus for application metrics monitoring. Prometheus is a monitoring tool that helps in analyzing the application metrics for flows and activities. Prometheus servers scrape data from the `HTTP /metrics` endpoint of the TIBCO FTL server. Prometheus integrates with Grafana, which provides better visual analytics.

For details, see "Prometheus Endpoints " in *TIBCO FTL® Administration*.

New tibMap APIs to remove multiple keys

Added new client APIs that allow applications to remove several map keys in one API call. In C API, the new calls are tibMap_RemoveMultiple and tibMap_RemoveMultipleWithLock.

Improved TIBCO FTL disk persistence

The new FTL 7.0 database has an improved ability to expand or compact large databases without interrupting clients. The new database also offers the potential for improved performance in some scenarios, such as when using high-latency disks. When upgrading from the 6.x version, the old 6.x database will be automatically imported into FTL 7.0. For details, see Migration With Disk Persistence section.

Audit Log for Authentication

When the TIBCO FTL server loglevel is set to "`auth:verbose`", TIBCO FTL server will log authentication results for incoming connections from clients or other TIBCO FTL servers.

Changes in Functionality

The following changes in functionality were introduced in recent releases of TIBCO FTL® - Enterprise Edition software:

7.0.0

Username and Password in Config File

Previously, if the FTL keystore existed (due to init-security or init-auth-only), and no satellite or dr connections were configured, FTL server could ignore the username/password in the FTL server yaml file. The keystore would be used for authentication instead. Now, username and password are no longer ignored. If username and password are invalid, the FTL servers may not form a quorum.

Change in Openssl Security Level

Previously, TLS connections accepted by the FTL REST API were configured with openssl security level 1. The default security level is now 2. The security level can be configured using the new configuration parameter "tls.security.level" in the FTL server yaml file.

Default Trust File for HTTPS Authenticators

Previously, if an https authenticator was configured (via "auth.url"), and no trust file was configured (via "auth.trust"), the default trust file was "ftl-auth-trust.pem". Now, the https authenticator will use the system trust store by default, unless "ftl-auth-trust.pem" actually exists.

Secure by Default without TLS

If FTL server is started with no pre-existing configuration (e.g., an empty data directory), and authentication is enabled, then by default the built-in FTL services will use secure transports. This includes the default persistence cluster, the default eFTL cluster, and the group service. If TLS is enabled, then this is the same as 6.x. However, if TLS is not enabled, then it will not be possible for 6.x FTL clients to use the built-in FTL services. If compatibility with 6.x clients is required, then either preserve the data directory on upgrade (which will preserve the pre-existing configuration), or set the FTL server configuration parameter "disable.default.security" to suppress the new behavior (when TLS is not enabled).

Authentication must also be configured if TLS is configured

If TLS is configured, then authentication must also be configured. Make sure that all the applications have correct roles.

Deprecated and Removed Features

The following tables list any features that have been deprecated or removed as of Release 7.0.1 of TIBCO FTL® - Enterprise Edition software:

For deprecated features, if relevant, useful alternatives to the deprecated features are listed. Any use of a deprecated feature should be discontinued as it may be removed in a future release. You should avoid becoming dependent on deprecated features and become familiar with the suggested alternative features.

Platforms

Affected Platform	Migration	Affected Release
Red Hat Enterprise Linux Server 7.x 64-bit on x86-64	Migrate to Red Hat Enterprise Linux Server 8.x or 9.x	7.0.1
Windows Server 2016	Migrate to Windows Server 2019 or Greater.	6.7.1
Windows Server 2012	Migrate to Windows Server 2016.	6.1.0
Windows Server 2008	Migrate to Windows Server 2016.	Deprecated in Release 5.3.0
Apple macOS on x86-64	Support on Apple macOS x86-64 is deprecated and will be removed in a future release	7.0.0
Apple macOS 10.14 64-bit, x86-64	Migrate to 10.15.x.	6.6.1
Apple macOS 10.13 64-bit, x86-64	Migrate to 10.14.x, 10.15.x.	6.6.0

Affected Platform	Migration	Affected Release
Apple macOS 10.12 64-bit, x86-64	Migrate to 10.14.x, 10.15.x.	6.1.0
Apple Mac OS X 10.11 64-bit, x86-64	Migrate to 10.14.x, 10.15.x	6.0.1
Red Hat Enterprise Linux Server 5.x 64-bit, x86-64	Migrate to 6.x or 7.x.	4.3.0
SUSE Linux Enterprise Server 11.x 64-bit, x86-64	Migrate to 12.x or 15.	6.7.0
SUSE Linux Enterprise Server 11.0 64-bit, x86-64	Migrate to 11.4 or 12.	4.2.0

Deprecated and Removed Features

Affected Component	Description	Deprecated Release	Removed Release
Monitoring Metrics	The following monitoring metric types are deprecated /** Queue message callback latency maximum */ #define TIB_MONITORING_TYPE_QUEUE_LATENCY_MSG_MAX 501	7.0.1	

Affected Component	Description	Deprecated Release	Removed Release
	#define TIB_MONITORING_TYPE_QUEUE_LATENCY_MSG_MAX_NAME “queue_latency_msg_max” /** Queue message callback latency mean - Deprecated */ #define TIB_MONITORING_TYPE_QUEUE_LATENCY_MSG_MEAN 502 #define TIB_MONITORING_TYPE_QUEUE_LATENCY_MSG_MEAN_NAME “queue_latency_msg_mean” /** Queue timer callback latency mean - Deprecated */ #define TIB_MONITORING_TYPE_QUEUE_LATENCY_TIMER_MEAN 506 #define TIB_MONITORING_TYPE_QUEUE_LATENCY_TIMER_MEAN_NAME “queue_latency_timer_mean” /** Queue timer callback latency standard deviation - Deprecated */ #define TIB_MONITORING_TYPE_QUEUE_LATENCY_TIMER_STDDEV 507 #define TIB_MONITORING_TYPE_QUEUE_LATENCY_TIMER_STDDEV_NAME “queue_latency_		

Affected Component	Description	Deprecated Release	Removed Release
	timer_stddev” /** Request/reply latency mean - Deprecated */ #define TIB_MONITORING_TYPE_ REQREPLY_LATENCY_MEAN 510 #define TIB_MONITORING_TYPE_ REQREPLY_LATENCY_MEAN_ NAME “reqreply_latency_mean” /** Request/reply latency standard deviation - Deprecated */ #define TIB_MONITORING_TYPE_ REQREPLY_LATENCY_STDDEV 511 #define TIB_MONITORING_TYPE_ REQREPLY_LATENCY_STDDEV_ NAME “reqreply_latency_stddev”		
tiblogsvc binary has been removed	The tiblogsvc binary has been removed.		7.0.0
TIB_REALM_PROPERTY_ LONG_TRUST_TYPE" API is deprecated	The `"TIB_REALM_PROPERTY_LONG_ TRUST_TYPE"` API is deprecated, along with `"TIB_REALM_HTTPS_ CONNECTION_TRUST_EVERYONE"`, `"TIB_REALM_HTTPS_CONNECTION_ USE_SPECIFIED_TRUST_FILE"`, and `"TIB_REALM_HTTPS_ CONNECTION_USE_SPECIFIED_ TRUST_STRING"`. Instead, use the one of the following APIs directly, or none of them: `"TIB_REALM_ PROPERTY_STRING_TRUST_FILE"`, `"TIB_REALM_PROPERTY_STRING_`	7.0.0	

Affected Component	Description	Deprecated Release	Removed Release
	TRUST_PEM_STRING", or "TIB_REALM_PROPERTY_BOOL_TRUST_EVERYONE"		
"custom.cert", "custom.cert.private.key", "custom.cert.private.key.password" are deprecated	FTL server yaml file configuration parameters "custom.cert", "custom.cert.private.key", "custom.cert.private.key.password" are deprecated. Use "tls.server.cert", "tls.server.private.key", and "tls.server.private.key.password" instead.	7.0.0	
Changing the matcher of an existing static durable	Previously, the feature to allow changing the matcher of an existing static durable was deprecated. It has now been removed.	6.10.2	7.0.0
FTL-guest role is deprecated	The "ftl-guest" role is deprecated. Use "ftl-admin" instead	7.0.0	
Use of "stdin" as a password	Use of "stdin" as a password specifier is deprecated. Use the plain password, file, or environment variable forms instead.	7.0.0	
Go API constant	The Go API constant PublisherSendPolicyBatching is deprecated. Use PublisherSendPolicyNonInline instead.	6.10.0	

Affected Component	Description	Deprecated Release	Removed Release
Display of Client Application Statistics	The display of client application statistics is removed.	6.10.0	6.10.0
FTL Monitoring Components	The FTL Monitoring component (monitoring directory) including Grafana and tibmongateway are removed. Use TIBCO® Messaging Monitor for TIBCO FTL®.	6.9.0	6.10.0
Monitoring Metrics	The following monitoring metric types are removed and data will not be returned. Use TIBCO® Messaging Monitor for TIBCO FTL®. #define TIB_MONITORING_TYPE_CONN_INFO 90001 #define TIB_MONITORING_TYPE_CONN_INFO_NAME "connection_definition" /** Transport connection - Number of matches performed at receiving side only*/ #define TIB_MONITORING_TYPE_CONN_RCVR_SIDE_MATCHES 90002 #define TIB_MONITORING_TYPE_CONN_RCVR_SIDE_MATCHES_NAME "receive_side_matches" /** Transport connection - Number of received messages that failed to match*/ #define TIB_MONITORING_TYPE_	6.9.0	6.10.0

Affected Component	Description	Deprecated Release	Removed Release
	CONN_RCVR_SIDE_DISCARDS 90003 #define TIB_MONITORING_TYPE_CONN_RCVR_SIDE_DISCARDS_NAME "failed_matches" /** Transport connection - Number of received messages matched the NULL matcher*/ #define TIB_MONITORING_TYPE_CONN_REC_NULL_MATCHES 90004 #define TIB_MONITORING_TYPE_CONN_REC_NULL_MATCHES_NAME "null_matches" /** Bytes sent. */ #define TIB_MONITORING_TYPE_CONN_SENT_BYTES 90005 #define TIB_MONITORING_TYPE_CONN_SENT_BYTES_NAME "connection_sent_bytes" /** Bytes received. */ #define TIB_MONITORING_TYPE_CONN_REC_BYTES 90006 #define TIB_MONITORING_TYPE_CONN_REC_BYTES_NAME "connection_received_bytes" /** Messages sent using normal matching procedure */ #define TIB_MONITORING_TYPE_		

Affected Component	Description	Deprecated Release	Removed Release
	CONN_SENT_MSGS_MATCHING 90007 #define TIB_MONITORING_TYPE_CONN_SENT_MSGS_MATCHING_NAME "connection_sent_msgs_match" /** Messages sent using without matching*/ #define TIB_MONITORING_TYPE_CONN_SENT_MSGS_OPAQUE 90008 #define TIB_MONITORING_TYPE_CONN_SENT_MSGS_OPAQUE_NAME "connection_sent_msgs_exp" /** Received messages that matched a certain matcher*/ #define TIB_MONITORING_TYPE_CONN_REC_MATCHING 90009 #define TIB_MONITORING_TYPE_CONN_REC_MATCHING_NAME "receive_matcher_matches" /** Sent messages that matched a certain matcher*/ #define TIB_MONITORING_TYPE_CONN_SENT_MATCHING 90010 #define TIB_MONITORING_TYPE_CONN_SENT_MATCHING_NAME "send_matcher_matches" /** End Point Description */		

Affected Component	Description	Deprecated Release	Removed Release
	#define TIB_MONITORING_TYPE_EP_INFO 90011 #define TIB_MONITORING_TYPE_EP_INFO_NAME "endpoint_description" /** End Point Delivered Messages */ #define TIB_MONITORING_TYPE_EP_DELIVERED_MSGS 90012 #define TIB_MONITORING_TYPE_EP_DELIVERED_MSGS_NAME "endpoint_delivered" /** End Point Published Messages */ #define TIB_MONITORING_TYPE_EP_PUBLISHED_MSGS 90013 #define TIB_MONITORING_TYPE_EP_PUBLISHED_MSGS_NAME "endpoint_published" /** Outgoing Messages that were discarded*/ #define TIB_MONITORING_TYPE_EP_SENT_NONMATCHING_MSGS 90014 #define TIB_MONITORING_TYPE_EP_SENT_NONMATCHING_MSGS_NAME "endpoint_out_discarded" /** Subscriber Delivered Messages */		

Affected Component	Description	Deprecated Release	Removed Release
	#define TIB_MONITORING_TYPE_SUB_DELIVERED_MSGS 90015 #define TIB_MONITORING_TYPE_SUB_DELIVERED_MSGS_NAME "subscriber_delivered" /** Publisher Published Messages */ #define TIB_MONITORING_TYPE_PUB_PUBLISHED_MSGS 90016 #define TIB_MONITORING_TYPE_PUB_PUBLISHED_MSGS_NAME "publisher_published" /** Outgoing Messages that were discarded*/ #define TIB_MONITORING_TYPE_PUB_SENT_NONMATCHING_MSGS 90017 #define TIB_MONITORING_TYPE_PUB_SENT_NONMATCHING_MSGS_NAME "publisher_discarded" /** Packets sent by connection. */ #define TIB_MONITORING_TYPE_CONN_PACKETS_SENT 90018 #define TIB_MONITORING_TYPE_CONN_PACKETS_SENT_NAME "connection_packets_sent" /** Packets received by connection. */		

Affected Component	Description	Deprecated Release	Removed Release
	#define TIB_MONITORING_TYPE_CONN_PACKETS_RECEIVED 90019 #define TIB_MONITORING_TYPE_CONN_PACKETS_RECEIVED_NAME "connection_packets_received" /** Packets retransmitted by connection. */ #define TIB_MONITORING_TYPE_CONN_PACKETS_RETRANSMITTED 90020 #define TIB_MONITORING_TYPE_CONN_PACKETS_RETRANSMITTED_NAME "connection_packets_retransmitted" /** Packets missed by connection. */ #define TIB_MONITORING_TYPE_CONN_PACKETS_MISSED 90021 #define TIB_MONITORING_TYPE_CONN_PACKETS_MISSED_NAME "connection_packets_missed" /** Packets lost outbound by connection. */ #define TIB_MONITORING_TYPE_CONN_PACKETS_LOST_OUTBOUND 90022 #define TIB_MONITORING_TYPE_CONN_PACKETS_LOST_		

Affected Component	Description	Deprecated Release	Removed Release
	OUTBOUND_NAME "connection_packets_lost_outbound" /** Packets lost inbound by connection. */ #define TIB_MONITORING_TYPE_CONN_PACKETS_LOST_INBOUND 90023 #define TIB_MONITORING_TYPE_CONN_PACKETS_LOST_INBOUND_NAME "connection_packets_lost_inbound" /** Multicast Receiver Stream Definition */ #define TIB_MONITORING_TYPE_MCAST_REC_INFO 90024 #define TIB_MONITORING_TYPE_MCAST_REC_INFO_NAME "mcast_receiver_stream_info" /** Multicast Sender Stream Definition */ #define TIB_MONITORING_TYPE_MCAST_SND_INFO 90025 #define TIB_MONITORING_TYPE_MCAST_SND_INFO_NAME "mcast_sender_stream_info" /** Multicast Receiver received bytes */ #define TIB_MONITORING_TYPE_MCAST_REC_BYTES 90026		

Affected Component	Description	Deprecated Release	Removed Release
	#define TIB_MONITORING_TYPE_ MCAST_REC_BYTES_NAME "mcast_receiver_bytes_received" /** Multicast Receiver delivered bytes */ #define TIB_MONITORING_TYPE_ MCAST_DEL_BYTES 90027 #define TIB_MONITORING_TYPE_ MCAST_DEL_BYTES_NAME "mcast_receiver_bytes_ delivered" /** Multicast Receiver received packets */ #define TIB_MONITORING_TYPE_ MCAST_REC_PACKETS 90028 #define TIB_MONITORING_TYPE_ MCAST_REC_PACKETS_NAME "mcast_receiver_packets_ received" /** Multicast Receiver delivered packets */ #define TIB_MONITORING_TYPE_ MCAST_DEL_PACKETS 90029 #define TIB_MONITORING_TYPE_ MCAST_DEL_PACKETS_NAME "mcast_receiver_packets_ delivered" /** Multicast receiver sent protocol packets*/ #define TIB_MONITORING_TYPE_		

Affected Component	Description	Deprecated Release	Removed Release
	MCAST_RCVR_SENT_PACKETS 90030 #define TIB_MONITORING_TYPE_MCAST_RCVR_SENT_PACKETS_NAME "mcast_receiver_packets_sent" /** Multicast sent NAK requests */ #define TIB_MONITORING_TYPE_MCAST_SENT_NAK_REQS 90031 #define TIB_MONITORING_TYPE_MCAST_SENT_NAK_REQS_NAME "mcast_receiver_nak_requests_sent" /** Multicast sent NAKs requested */ #define TIB_MONITORING_TYPE_MCAST_SENT_NAKS 90032 #define TIB_MONITORING_TYPE_MCAST_SENT_NAKS_NAME "mcast_receiver_naks_requested" /** Multicast Receiver lost packets */ #define TIB_MONITORING_TYPE_MCAST_LOST_PACKETS 90033 #define TIB_MONITORING_TYPE_MCAST_LOST_PACKETS_NAME "mcast_receiver_lost_packets" /** Multicast Receiver duplicate		

Affected Component	Description	Deprecated Release	Removed Release
	packets received*/ #define TIB_MONITORING_TYPE_MCAST_DUP_REC_PACKETS 90034 #define TIB_MONITORING_TYPE_MCAST_DUP_REC_PACKETS_NAME "mcast_receiver_duplicates_received" /** Multicast Receiver data errors*/ #define TIB_MONITORING_TYPE_MCAST_DATA_ERRORS 90035 #define TIB_MONITORING_TYPE_MCAST_DATA_ERRORS_NAME "mcast_receiver_data_errors" /** Packets sent by multicast sender. */ #define TIB_MONITORING_TYPE_MCAST_SENDER_PACKETS_SENT 90036 #define TIB_MONITORING_TYPE_MCAST_SENDER_PACKETS_SENT_NAME "mcast_sender_packets_sent" /** Bytes sent by multicast sender. */ #define TIB_MONITORING_TYPE_MCAST_SENDER_BYTES_SENT 90037 #define TIB_MONITORING_TYPE_		

Affected Component	Description	Deprecated Release	Removed Release
	MCAST_SENDER_BYTES_SENT_ NAME "mcast_sender_bytes_ sent" /** Packets retransmitted by the multicast sender. */ #define TIB_MONITORING_TYPE_ MCAST_SENDER_PACKETS_ RETRANSMITTED 90038 #define TIB_MONITORING_TYPE_ MCAST_SENDER_PACKETS_ RETRANSMITTED_NAME "mcast_ sender_packets_retransmitted" /** Packets lost outbound by multicast sender. */ #define TIB_MONITORING_TYPE_ MCAST_SENDER_PACKETS_ LOST_OUTBOUND 90039 #define TIB_MONITORING_TYPE_ MCAST_SENDER_PACKETS_ LOST_OUTBOUND_NAME "mcast_sender_packets_lost" /** Backlog highest size during the previous period*/ #define TIB_MONITORING_TYPE_ SEND_BACKLOG_MAX_SIZE 90040 #define TIB_MONITORING_TYPE_ SEND_BACKLOG_MAX_SIZE_ NAME "send_backlog_maximum_ size"		
User Interface, Statistics Page	The display of client application	6.9.1	6.9.1

Affected Component	Description	Deprecated Release	Removed Release
(FTL-11958)	statistics is now removed.		
Server Clusters	A cluster of seven servers in a quorum is no longer supported. Three or five servers are recommended.	6.9.0	6.9.0
TLS without Authentication	TLS without Authentication is deprecated.	6.8.0	
Pre-Built Docker Images	Pre-built Docker images are no longer supplied with FTL software distributions.	6.7.1	6.7.1
Administrative GUI	The display of client application statistics is deprecated. It is planned for removal in the next minor release.	6.7.1	6.10.0
Persistence Service	The `disk_mode` configuration parameter for a persistence cluster in the web API is deprecated. Instead, use `disk_swap` and/or `disk_persistence`.	6.7.0	
FTL Server Monitoring	For the web API call `api/v1/persistence/ftlservers/<ftl server name>/status` and in the administrative GUI, FTL Server status, the following fields are deprecated: • `current_connection`	6.7.0	

Affected Component	Description	Deprecated Release	Removed Release
	- `max_connections` - `rejected_connections` - `lookup_failures` Also in the administrative GUI, the **`Connections at this server`** section is deprecated.		

Persistence Monitoring Web API	REST API calls to locations that started with
	`api/v1/persistence`
	now begin instead with
	`api/v1 /persistence/clusters`
	Until (but not including) the removal release, the old locations automatically map to the new calls.
	Note: Update your code appropriately.
		6.1.0	

Go API	`MsgContent` is deprecated. Instead, use methods of the `Message` object to marshal and unmarshal data between messages and Go structs.	6.1.0	
FTL Server Web API	The REST API call `server {"cmd":"shutdown"}` is obsolete. Instead, use `ftlservers`	6.1.0	6.1.0

Affected Component	Description	Deprecated Release	Removed Release
	`{"cmd":"shutdown"}`		
FTL Server Administration Utility	Support for reading parameters from a configuration file is deprecated. Supply all commands and parameters on the command line.	6.0.0	6.0.0
Agent	The agent component is obsolete. FTL clients and servers running in Docker containers no longer require the agent.	6.0.0	6.0.0
Prometheus	Support for Prometheus is obsolete.	5.4.0	6.0.0
Monitoring Message Stream	New monitoring message types replace old types, which are deprecated: • 90012 replaces type 2. • 90013 replaces type 1. Use the new types. The old types remain in Release 5.4.0 for backward compatibility, but will become obsolete in a future release.	5.4.0	
Bridge Setup Scripts	The Python scripts `init_bridge.py` and `init_dtcp_bridge.py` are obsolete. Use the realm server web API instead.	5.2.0	5.2.0
Realm Configuration Python Scripts	The realm configuration Python script, `rs_script.py`, and its supporting utility scripts are	5.2.0	5.2.0

Affected Component	Description	Deprecated Release	Removed Release
	obsolete. Use the realm server web API instead.		
Group Setup	The group facility is automatically enabled. The Python script `init_groups.py` is obsolete and no longer needed. The web API calls `POST realm/groupserver` and `DELETE realm/groupserver` are obsolete and no longer needed.	5.2.0	5.2.0
Realm Server Internal JAAS	The realm server now relies on a separate authentication service, rather than an internal JAAS component.	5.2.0	5.2.0
Realm Server Monitoring Interface	The realm server no longer stores historical monitoring data. For replacement functionality, see *TIBCO FTL Monitoring*.	5.0.0	5.0.0
Edit Transport Configuration Manually	The realm server GUI transport definition page no longer support manually editing a transport's JSON definition. To modify the transport definition, use either the GUI or the web API. See "PUT realm/transports/<name>" in TIBCO FTL Administration.	5.0.0	5.0.0
Adapter	The adapter converts and	4.2.0	4.3.0

Affected Component	Description	Deprecated Release	Removed Release
	forwards messages between TIBCO FTL and TIBCO Rendezvous. This component is obsolete. This functionality is now part of TIBCO Rendezvous Network Server software.		
API	API calls that facilitated request/reply interactions between TIBCO FTL and TIBCO eFTL are obsolete.	4.2.0	Deactivated in 4.2.0. Removed in 5.0.0.
FTL Server Configuration File	Support for eFTL Service configuration parameters `server.cert`, `private.key`, and `private.key.password` is deprecated. Instead, use FTL Server configuration parameters `custom.cert`, `custom.cert.private.key` , and `custom.cert.private.key.password` .	6.2.0	November 2019

Migration and Compatibility for 7.0.0

The following information provides migration procedures for this release of TIBCO FTL® - Enterprise Edition 7.0.1.

Compatibility with Earlier Releases of TIBCO FTL

⚠ **Warning:** TIBCO FTL® - Enterprise Edition 7.0.0 data on disk representation is not compatible with an earlier version of TIBCO FTL® - Enterprise Edition. Hence, before upgrading to TIBCO FTL® - Enterprise Edition 7.0.0, see Downgrading to 6.10.x Release of FTL and Downgrading to 6.9.x Release of FTL section as applicable to take backups.

Upgrading or Migrating to a New Release

Read these instructions before upgrading from an earlier TIBCO FTL release.

The upgrade or migration tasks you must complete depend upon the release from which you are upgrading.

Upgrading from Release 6.x

If you are upgrading from a release earlier than TIBCO FTL Release 6.7.1, you must first upgrade to Release 6.10.1 and then follow instructions for upgrading from 6.7.1 or later to TIBCO FTL 7.0.0.

You can upgrade to TIBCO FTL Release 7.0.0 directly from TIBCO FTL release 6.7.1 or later.

This procedure uses a rolling upgrade, which is to upgrade one server at a time. This lets you upgrade a network without a total service interruption.

Upgrading from Release 6.7.1 or Later

This procedure uses a rolling upgrade, which is to upgrade one server at a time. This lets

you upgrade a network without a total service interruption.

This procedure enables you to upgrade to TIBCO FTL Release 7.0.1 from TIBCO FTL release 6.7.1 or later.

Procedure

1. Determine which core servers are leaders, and plan the order in which you want to upgrade host computers to the new release of TIBCO FTL. A recommended sequence is as follows:

   a. disaster recovery location, non-leader core servers
   b. disaster recovery location, leader core server
   c. disaster recovery location, auxiliary server

d. primary location, non-leader core servers
e. primary location, leader core server
f. primary location, auxiliary server
g. satellite location, non-leader core servers
h. satellite location, leader core server
i. satellite location, auxiliary server

2. Check the persistence clusters status table and its services list sub-table to verify that all the persistence services in the cluster are a) running, b) part of the quorum, and c) up to date.

   See Persistence Clusters Status Table , TIBCO FTL Servers Status Page, and Servers List.

3. Address the first or next host computer in your plan and stop its TIBCO FTL server process.

   This stops all services under that TIBCO FTL server automatically.

4. Uninstall the old TIBCO FTL installation package from that server's host computer. See instructions in *TIBCO FTL Installation*.

5. Install the new release of the full TIBCO FTL product on that host computer. See instructions in *TIBCO FTL Installation*.

6. Ensure that all server computers in the cluster have their clocks synchronized.

7. Repeat steps 2-6 for the next TIBCO FTL server in your plan. Continue for each server until you have upgraded all the TIBCO FTL servers.

8. Upgrade all application clients.

Migration With Disk Persistence

Migrating TIBCO FTL Servers to TIBCO FTL Release 7.0.0 from TIBCO FTL Release 6.7.1 or Later 6.x.x versions

TIBCO FTL Server 7.0.1 disk persistence database is different from TIBCO FTL server 6.x.x disk persistence database, hence when you upgrade to TIBCO FTL server 7.0.1, a downgrade to an older version is not possible without the TIBCO FTLServer data directory backup from the earlier version.

> **Note:** During upgrade to TIBCO FTL release 7.0, TIBCO FTL server must make a copy of all pending data into the new database. Therefore disk space requirements temporarily double during migration. For example, if the data directory for TIBCO FTL 6.x.x was 100 GB, ensure that at least 200 GB is provisioned during upgrade.

Follow the procedure in Upgrading from Release 6.x

Once the upgrade is complete, the old database can be moved or deleted.The old database files will have the suffix ".imported" appended to the file name.

Migrating TIBCO FTL Servers to TIBCO FTL Release 7.0.1 from TIBCO FTL Release 6.7.1 or Later 6.x.x Versions When Provisioned Disk Space Cannot be Expanded

Determine which TIBCO FTL servers are leaders, and plan the order in which you want to upgrade host computers to the new release of TIBCO FTL.

Suppose ftls1 is the leader of the persistence cluster, and ftls2 and ftls3 are followers. This is the recommended sequence

a. Shutdown one of the servers which is the follower (ftls3).

b. Delete the data directory of this server (ftls3) that was just shutdown.

c. Start a TIBCO FTL 7.0.1 server and let it catch up from the other two servers from 6.x.

d. Shutdown another one of the 6.x TIBCO FTL servers which is also a follower (ftls2).

e. Delete the data directory of this server (ftls2) that was just shutdown.

f. Start a TIBCO FTL 7.0.1 server and let it catch up from the other two servers.

g. Once the two 7.0.1 TIBCO FTL servers are fully in sync, shutdown the 6.x leader (ftls1) TIBCO FTL server.

h. Delete the data directory of this TIBCO FTL server (ftls1).

i. Start a TIBCO FTL 7.0.1 server and let it catch up from the other two servers already from 7.0.1.

Migrating TIBCO FTL servers from one data center to another

When you are migrating from TIBCO FTL release 6.7.1 or later where disk persistence is enabled, for data center migration or upgrades, you can shut down the servers and copy the disk persistence related database files for data center migration use cases. You must shut down all servers before copying these files.

Enabling Disk Persistence for the First Time

When migrating from in-memory persistence to disk-based persistence, two rolling upgrades are required. First, upgrade to 6.7.0 (or later), enable disk persistence or disk swap, then restart the servers again.

Migrating to a Different Host

You can migrate an TIBCO FTL server installation to a replacement host computer. Complete the following procedure.

Procedure

1. Install the full TIBCO FTL product on the new host computer. See instructions in *TIBCO FTL Installation*.
2. On the new host computer, ensure that the TIBCO FTL server is not running.
3. On the old host computer, shut down the TIBCO FTL server.
4. Copy the data directory for the realm service and, if using disk persistence, the persistence service(s) run by the server.
5. Copy any security-related files that were distributed to the server as part of the --init-security or --init-auth-only procedure (i.e., the `ftl-tport.p12` and `ftl-trust.pem` files).
6. Remap DNS so that any FTL clients or FTL servers that are currently running are able to reconnect to the new host computer.
7. Start the TIBCO FTL server on the new host computer.

Eliminating the TIBCO FTL Keystore (Authentication Only)

For TIBCO FTL 6.x configurations that used authentication, but did not use TLS, you had to generate an TIBCO FTL keystore by using `tibftlserver --init-auth-only`. Then, you had to distribute the TIBCO FTL keystore and trust files to the data directory of each TIBCO FTL server.

To continue running TIBCO FTL as you did for TIBCO FTL 6.x, do not take any action after upgrading.

When using version TIBCO FTL 7.x, the TIBCO FTL keystore is no longer necessary for authentication. Users that want to enable oauth2 authentication may optionally eliminate the TIBCO FTL keystore after upgrading to TIBCO FTL 7.x. For example, this will allow TIBCO FTL server to enforce oauth2 token expirations.

To eliminate the TIBCO FTL keystore, you must follow this procedure because TIBCO FTL servers that have the TIBCO FTL keystore cannot communicate with TIBCO FTL servers that do not have the TIBCO FTL keystore. This procedure requires a period of time where all TIBCO FTL servers are shut down.

Procedure

1. Upgrade all TIBCO FTL servers to TIBCO FTL 7.x. For more information, see Upgrading from Release 6.x

2. Upgrade all TIBCO FTL clients to TIBCO FTL 7.x. For more information, see Upgrading from Release 6.x

3. Save the state of all in-memory persistence clusters to preserve pending messages. For more information, see Saving and Loading Persistence State. If all persistence clusters use disk persistence, no action is needed.

4. Shut down all TIBCO FTL servers, including TIBCO FTL servers at satellite or DR sites.

5. For each TIBCO FTL server, remove `ftl-tport.p12` and `ftl-trust.pem` from the data directory of the server.

6. Restart all TIBCO FTL servers. Clients reconnect automatically.

Eliminating FTL-Generated Certificates (Authentication and TLS)

For TIBCO FTL 6.x configurations that used both authentication and TLS, you had to generate TIBCO FTL certificates by using `tibftlserver --init-security`. Then, you had to distribute the TIBCO FTL keystore and trust files to the data directory of each TIBCO FTL server.

To continue running TIBCO FTL as you did for TIBCO FTL 6.x, do not take any action after upgrading.

When using version TIBCO FTL 7.x, FTL-generated certificates are no longer necessary for TLS. Users that want to control TLS certificates or enable oauth2 authentication may optionally eliminate FTL-generated certificates after upgrading to TIBCO FTL 7.x. For example, this will allow TIBCO FTL server to enforce oauth2 token expirations.

However, to eliminate FTL-generated certificates, and provide their own certificates, you must follow the procedure in this section because TIBCO FTL servers that use FTL-generated certificates cannot communicate with TIBCO FTL servers that do not use the FTL-generated certificates. This procedure requires a period of time where all TIBCO FTL servers are shut down. Also note that when using user-defined certificates, secure peer-to-peer transports are not permitted. Only secure server-based transports are permitted (for example, persistence service or group service transports).

Procedure

1. Ensure that no client applications are using secure peer-to-peer transports.
2. Upgrade all TIBCO FTL servers to TIBCO FTL 7.x. For more information, see Upgrading from Release 6.x
3. Upgrade all TIBCO FTL clients to TIBCO FTL 7.x. For more information, see Upgrading from Release 6.x.When you restart the TIBCO FTL client at version 7.x, provide the trust certificates that correspond to the user-defined certificates that you plan to use later.
    a. If the trust certificates are installed in the system trust store, install them before restarting the client.
    b. If the trust certificates are passed to the client API as a PEM file, concatenate the trust certificates with the FTL-generated trust file (`ftl-trust.pem`). Pass the

resulting combined PEM file to the client API.

4. Save the state of all in-memory persistence clusters to preserve pending messages. For more information, see Configuring Persistence. If all persistence clusters use disk persistence, no action is needed.
5. Shut down all TIBCO FTL servers, including TIBCO FTL servers at satellite or DR sites.
6. For each TIBCO FTL server, remove `ftl-tport.p12` and `ftl-trust.pem` from the data directory of the server. Make the following changes to the TIBCO FTL server yaml configuration file.
    a. Remove `tls.secure`.
    b. Add the user-defined certificates (`tls.server.cert`, `tls.server.private.key`, `tls.server.private.key.password`). Ensure that each certificate is appropriate for the specific TIBCO FTL server's hostname.
    c. Add the trust certificates corresponding to the user-defined certificates (`tls.client.trust.file`). Alternatively, install them in the system trust store. For more information, see the Enabling TLS for TIBCO FTL Server
7. Restart all TIBCO FTL servers. Clients reconnect automatically by using the trust information provided earlier.

Downgrading to 6.10.x and 6.9.x Release of FTL TIBCO FTL® - Enterprise Edition

The following information provides downgrading procedures for this release of TIBCO FTL® - Enterprise Edition7.0.1.

Downgrading to 6.10.x Release of FTL

TIBCO FTL® - Enterprise Edition 7.0.0 data on disk representation is not compatible with an older version of TIBCO FTL® - Enterprise Edition. Hence, before upgrading to TIBCO FTL® - Enterprise Edition 7.0.0, use this procedure to back up the existing TIBCO FTL® - Enterprise Edition 6.10.x database. Use this procedure to downgrade to TIBCO FTL® - Enterprise Edition 6.10.x.

Steps to be taken before upgrading to TIBCO FTL® - Enterprise Edition 7.0.0

1. Back up the realm according to the documentation from TIBCO FTL® 6.10.x release
   a. For example, `tibftladmin --backup_realm -ftls <host:port>` (`host:port` of one of the FTLServers from the FTLServer cluster)
2. Back up the persistence clusters according to the TIBCO FTL® - Enterprise Edition 6.10.x documentation. Use these steps when disk persistence is enabled for `ftl.default.cluster` and other user-defined clusters.
   a. `tibftladmin --backup_persist --cluster ftl.default.cluster -ftls <host:port>` (host:port of one of the servers from the FTLServer cluster)

> **Note:** Check the progress of the backup by using the REST API, `/api/v1/persistence/ftl.default.cluster/servers`, look for the 'backup_in_progress' field from the status JSON and if complete the field value is false

b. If you are running TIBCO Enterprise Message Service™ as a part of the FTLServer cluster, then run `tibftladmin --backup_persist --cluster _embedded_tibemsd -ftls <host:port>` (host:port of one of the FTLservers from the FTLServer cluster)

> **Note:** Check the progress of the back up by using the REST API, `/api/v1/persistence/_embedded_tibemsd/servers`, look for the 'backup_in_progress' field from the status JSON and if complete the field value is false

c. Also if you have any user-defined persistence clusters, repeat this process for each user-defined persistence clusters

> **Note:** Check the progress of the backup using the REST API

d.

> **Note:** If disk persistence is not enabled, see TIBCO FTL® - Enterprise Edition documentation for Suspending a Persistence Cluster and Saving the State of a Persistence Service

3. Move the generated realm backups directory from the data directory of the realm as specified in the YAML file to some known location. For example, if the data directory for the realm is

   a. `TIBCO_HOME/disk-persistence/ftlserver1/realm/data`, then the backups directory would be in `TIBCO_HOME/disk-persistence/ftlserver1/realm/data/backups`, move this backups directory to some known location. For example, `$HOME/realm/backups`

4. Move the generated default cluster backup files from the persistence cluster's data directory to some well-defined location. For example, if the data directory for the persistence cluster is set to:

   a. `TIBCO_HOME/disk-persistence/ftlserver1/persist/data`, then the backup files are named something like this in the same directory
   b. `default_ftlserver1_2023-07-28_10-24-40-567.persist.backup`
   c. Move this file to some well-defined location. For example, move it to `$HOME/persist/backups`

5. If you are running EMS as a part of the FTLServer cluster, similarly move the backup files generated in 2.1.2, related to the `_embedded_tibemsd` cluster to some known location. For example, move the files to `$HOME/persist/backups`. The files related to the `_embedded_tibemsd` cluster are in the path specified in the YAML file for tibemsd `-store` YAML option

6. If you have any user-defined persistence clusters, move the backup files to some known location.

7. You can now upgrade from 6.10.x to 7.0.0

8. You can downgrade to TIBCO FTL® - Enterprise Edition 6.10.x, if required
   a. Gracefully shut down all the TIBCO FTL® - Enterprise Edition 7.0.0 FTLServers
   b. Restore the state from the **backups** for the realm, the **backup files** associated with **ftl.default.cluster** and **_emedded_tibemsd** cluster. But before restoring, run the following steps:
      i. Delete the content of the realm data directory for all the three FTL servers in the cluster
      ii. Delete the content of the persistence cluster data directory for the `ftl.default.cluster` for all the three FTL servers in the cluster
      iii. Delete the content of the persistence cluster data directory for the _ `emedded_tibemsd cluster` for all the three FTL servers in the cluster
      iv. Delete the data directory content for any user-defined persistence clusters

      > **Note:** Any data that was sent or durables created after the upgrade would be lost at this point. The state is restored from the backups taken earlier than the upgrade. Similarly any deployments made after the upgrade would be lost

   c. Run the `tibftladmin` command to restore the realm data from the backups directory (assuming `$HOME/realm/backups` is where you saved off the realm backups earlier)

i. `tibftladmin --restore_realm --backupdir $HOME/realm/backups --datadir TIBCO_HOME/disk-persistence/ftlserver1/realm/data --name ftlserver1`

d. Run the `tibftladmin` command to restore the persistence cluster related files. Example for the default cluster related restore is

i. `tibftladmin --restore_persist --backupdir $HOME/persist/backups --datadir TIBCO_HOME/disk-persistence/ftlserver1/persist/data --name default_ftlserver1`

e. Similarly run the `–restore_persist` command for the _embedded_tibemsd cluster. Example is

i. `tibftladmin --restore_persist --backupdir $HOME/persist/backups/ --datadir /opt/deployment/ftlserver1/ftlstore_data/ --name ftlserver1.`

f. Also if you have any user-defined persistence clusters, repeat this process for each of the user-defined persistence clusters

> **Note:** If disk persistence is not enabled for `ftl.default.cluster` and other user-defined clusters, see TIBCO FTL® - Enterprise Edition documentation on restoring state

9. Now restart all the FTLServers from previous version 6.10.0

   a. Ensure that the durables and messages earlier than the upgrade are preserved

   b. If you are running EMS, make sure that the EMS related topics, queues, and messages before the upgrade are preserved

Downgrading to 6.9.x Release of FTL

TIBCO FTL® - Enterprise Edition 7.0.0 data on disk representation is not compatible with an older version of TIBCO FTL® - Enterprise Edition. Hence, before upgrading to TIBCO FTL® - Enterprise Edition 7.0.0, use this procedure to back up the existing TIBCO FTL® - Enterprise Edition 6.9.x database. Use this procedure to downgrade to TIBCO FTL® - Enterprise Edition 6.9.x.

Steps to be taken before upgrading to TIBCO FTL® - Enterprise Edition 7.0.0

1. Back up the realm according to the documentation from TIBCO FTL® 6.9.x release
    a. For example, `tibftladmin --backup_database -ftls <host:port>` (`host:port` of one of the FTLServers from the FTLServer cluster)
2. Back up the persistence clusters according to the TIBCO FTL® - Enterprise Edition 6.9.x documentation. Use these steps when disk persistence is enabled for `ftl.default.cluster` and other user-defined clusters
    a. Go to the FTLServer UI that shows the running persistence clusters. Click the **backup** icon for
        i. `ftl.default` cluster
        ii. Any user-defined persistence clusters
        iii. `_embedded_tibemsd` cluster
        iv. Look for the FTLServer log message that indicates that the backup is complete. For example, look for 'Finished backup of'

    > **Note:** If disk persistence is not enabled, see the TIBCO FTL® - Enterprise Edition documentation for Saving the State of a Persistence Service and Suspending a Persistence Cluster

3. Move the realm backups directory, from the realm data directory as specified in the YAML file, to some known location. For example, if the data directory for the realm is
    a. `TIBCO_HOME/disk-persistence/ftlserver1/realm/data`, then the backups directory would be in `TIBCO_HOME/disk-persistence/ftlserver1/realm/data/backups`, move this backups directory to some known location. For example, `$HOME/realm/backups`
4. Move the cluster backup files from the persistence cluster's data directory to a well-defined location. You can do it for one of the servers from the cluster. You do not need to save the files from all the servers from the FTLServer cluster. For example, if the data directory for the persistence cluster is set to
    a. `TIBCO_HOME/disk-persistence/ftlserver1/persist/data`, then the backup files are named something like this in the same directory
    b. `default_ftlserver1_2023-07-28_10-24-40-567.persist.backup`

   c. Move this file to some well-defined location. For example, move it to `$HOME/persist/backups`

5. If you are running EMS as part of the FTLServer cluster, move the embedded cluster related backup file to the known location. For example:

   a. `ftlserver1_2023-07-31_15-29-15-448.persist.backup` is the backup file of the embedded `tibemsd` cluster. The FTLserver is named `ftlserver1`. Hence the backup is named according to the FTLserver name

6. If you are running user-defined persistence clusters, move the backup files related to the user-defined clusters to a known location

7. You can now upgrade from 6.9.x to 7.0.0

8. You can downgrade to TIBCO FTL® - Enterprise Edition 6.9.x, if required:

   a. Gracefully shut down all the TIBCO FTL® - Enterprise Edition 7.0.0 FTLServers

   b. Before restoring the state from backup, run the following steps:

      i. Delete the content of the realm data directory for all the three FTL servers in the cluster

      ii. Delete the content of the persistence cluster data directory for the `ftl.default.cluster` for all the three FTL servers in the cluster

      iii. Delete the contents of the persistence cluster data directory for the _ `emedded_tibemsd` cluster for all the three FTL servers in the cluster

      iv. Delete the content of the data directory of user-defined clusters

      > **Note:**
      > Any data that was sent or durables created after the upgrade would be lost at this point. The state is restored from the backups taken earlier than the upgrade. Similarly any deployments made after the upgrade would be lost

      > **Note:**
      > If disk persistence is not enabled for `ftl.default.cluster` and other user-defined clusters, see TIBCO FTL® - Enterprise Edition documentation on restoring state

   c. Restore the state from the backups for the realm using the procedure in the

TIBCO FTL® - Enterprise Edition 6.9.x documentation. The procedure is a manual process of renaming the files from the backups directory and copying them to the realm 'data' directory. For example

i. `mv config_ftlserver1_2023-07-31_15-31-07-026.persist.backup config_ftlserver1.persist`

ii. `mv rs_ftlserver1_2023-07-31_15-31-07-026.dat.backup rs_ftlserver1.dat`

d. Rename the backup file of `ftl.default.cluster` from `.backup` to `.persist`. For example

`mv default_ftlserver1_2023-07-31_15-27-46-406.persist.backup default_ftlserver1.persist`. This must be moved to the data directory of the `ftl.default.cluster`

e. Similarly move the backup of `embedded_tibemsd` from `.backup` to `.persist`

```
mv
ftlserver1_2023-07-31_15-29-15-448.persist.backup ftlserver1.persist
```

f. Similarly if you have any user-defined persistence clusters and their backup files, move them to the data directory of the user-defined persistence cluster

9. Restart all the FTLServers from the previous version of TIBCO FTL® - Enterprise Edition 6.9.x

   a. Ensure that the durables and messages earlier than the upgrade are preserved

   b. If you are running EMS, make sure that EMS related topics, queues, and messages earlier than the upgrade are preserved

Closed Issues

The following issues have been fixed in this release of TIBCO FTL® - Enterprise Edition.

7.0.1

Key	Summary
FTL-15058	FTL Server UI erroneously reports "Incorrect username or password" when accessing different FTLServers via the FTLServer UI and FTL Server is setup with user defined certs.
FTL-15055	Fixed a FTLServer UI defect where in the FTLServer UI incorrectly displays ‘Login as anyone’ page when accessing different FTLservers from the same cluster via the FTLServer UI.
FTL-15030	Fixed an issue where UI browser sessions were shared between primary, satellite, and DR servers when running on the same host.
FTL-15001	Fixed an issue where FTL server could sometimes drop HTTP requests with large request bodies.
FTL-14977	Fixed an issue where an FTL client's DTCP transport might not be able to establish a connection after the client reconnects to FTL server following a period where a majority of FTL servers are down.
FTL-14976	Fixed an issue where the quorum would reform at the active site when the transport to the dr site was interrupted. This caused a short interruption of messaging at the active site. Note that messaging is expected to proceed normally at the active site even if the dr site is temporarily unavailable.
FTL-14971	Fixed an issue where the primary FTL server might report that a realm deployment is complete before the deployment actually completed at a satellite site. This issue could affect users of Active Spaces who enable mirroring.

Key	Summary
FTL-14969	Fixed an issue where, after restarting the FTL server process, the FTL server would occasionally fail to initialize with error "Dynamic durable destroy request timed out".
FTL-14966	Fixed an issue where, in configurations that use FTL-generated certificates, FTL server might generate a certificate with a negative serial number. Note that, by default, golang now rejects certificates with negative serial numbers.
FTL-14958	Fixed an issue where, on Windows, inline event queues that dispatch secure TCP-based transports could sometimes stop receiving messages. This issue could also lead to disruptions of the FTL server quorum or persistence quorums when using secure transports.
FTL-14947	LDAP performs case-insensitive lookups properly.
FTL-14946	Fixed an issue on Windows where, if mTLS authentication is enabled, FTL server could truncate responses to REST requests. When this occurs, the REST client sees an early EOF.
FTL-14934	Fixed an issue where the persistence quorum might take longer than intended to reform after a network interruption on the cluster transport.
FTL-14910	Fixed an issue where FTL servers using OAuth2 authentication for the FTL UI could crash.
FTL-14906	Added warnings for common errors involving connections to or between FTL servers.
FTL-14899	Fixed a performance issue where message delivery from the persistence service is slow for a standard durable with no prefetch. The issue occurred in this scenario.
FTL-14888	Realm Services UI correctly shows whether server has TLS enabled.
FTL-14884	Fixed an issue where FTL server would open an unused port on localhost in some situations.

Key	Summary
FTL-14856	Added sample configuration for eFTL with OAuth2 on EMS channels .
FTL-14750	Fixed an issue where it was not possible to enable message tracing for a running client through the REST API or user interface.

7.0.0

Key	Summary
FTL-14762	Fixed an issue where, in configurations using a direct path transport backed by a standard durable without prefetch, messages could occasionally be delivered out of order.
FTL-14700	If an application creates a subscriber on an endpoint with a store and a direct path transport, and then closes the subscriber and the realm, the FTL library could leak memory.
FTL-14696	Previously, the flat-file auth provider required a url of the form "file://<path>", which did not work for absolute paths on Windows. FTL server now accepts "file:<path>" as a valid specification for the flat-file auth provider.
FTL-14578	Fixed an issue where, in rare cases, the persistence service might leak state related to a closed publisher. This could cause messages to re-appear after being unacked.
FTL-14569	Fixed an issue where applications that use routed stores could not receive realm configuration updates after multiple reconnects to FTL server.
FTL-14540	In rare cases, the persistence service could crash when a client uses the SendRequest/SendReply API during a quorum reformation.
FTL-14482	Fixed an issue where, if a persistence follower loses contact with the other quorum members due to a network partition, the follower could end up taking leadership when the partition is fixed.
FTL-14470	Improved warnings regarding stalls when writing to disk

Key	Summary
FTL-14469	Clarified the error message that is generated when an FTL application requests a lock without a lock retry duration, and the lock cannot be immediately granted to the application.
FTL-14468	tibftlserver may restart a given service even after it has exhausted the max number of restarts.
FTL-14466	Restarting a sending application that uses an endpoint configured with multicast transport can cause message loss, this is especially true if the multicast transport is configured with a range of ports.
FTL-14450	Fixed an issue where FTL could not be started with an even number of core servers.
FTL-14411	Fixed an issue where FTL might inadvertently create a duplicate connection on a route, causing reconnects on the route and delays in message delivery.
FTL-14407	Persistent publishers leaked by the application are now garbage collected when the application closes the realm.
FTL-14389	Clarified the error message that occurs when an application attempts to remove a key from a persistent map while disconnected from the persistence cluster.
FTL-14367	Fixed an issue where map remove calls could stall when disk persistence is enabled. For example this could affect EMS server running with FTL stores and compaction enabled.
FTL-14333	Fixed an issue where, if disk persistence is async, the tibMap_RemoveMultiple API could cause incorrect behavior, such as missing or incorrect values. This can happen if a set or remove call for one of the affected keys is made shortly after the tibMap_RemoveMultiple call returns.
FTL-14332	When 2 out of 3 FTL servers are started, the default persistence cluster should form a quorum after a delay. Fixed an issue that could prevent the default quorum from forming.
FTL-14331	Fixed an issue where, if disk persistence and message ttl are enabled, and there

Key	Summary
	are many messages to expire, the persistence service could exit following a database error.
FTL-14310	Fixed an issue where, if a subscriber on a standard durable (no prefetch) was running, and its dynamic durable was administratively deleted, the persistence service could report an exception "Bad range used" and stop functioning.
FTL-14308	Fixed an issue where, if FTL stores are used for EMS, EMS compaction might cause publishers to stall.
FTL-14298	tibRealm_Close API may leak resources when called with an exception set.
FTL-14284	Fixed an issue where routes might stall, or take a long time to fail over, if subscriptions are being created and destroyed rapidly.
FTL-14282	Fixed an issue where, after a failover to a new persistence cluster leader, the redelivery limit on a durable might be enforced prematurely. A message could be discarded after being delivered one fewer times than max_delivery.
FTL-14280	Fixed an issue where, in rare cases, if a persistence service has to sync from the leader, the sync could fail with exception "Invalid system property block size"
FTL-14271	Fixed an issue where the persistence service could crash on shutdown if routed stores are in use.
FTL-14267	Improved the error message that would occur when, during a DR failback procedure, a DR FTL server was started at the original primary site without cleaning up the old primary's database.
FTL-14259	FTL server now exits if it is unable to bind the FTL server port.
FTL-14224	Fixed an issue where, in environments with very high network and/or disk latencies, FTL may not be able to establish routes between persistence clusters in a forwarding zone.
FTL-14220	Fixed an issue where, in request-reply patterns over a route, it was possible for the reply to be dropped. The request message was sometimes delivered before

Key	Summary
	interest in the reply could propagate, leading to a drop of the reply.
FTL-14187	Fixed handling of the case where a static durable is deleted from the configuration, and then a new static durable of the same name, but different matcher, is created.
FTL-14177	Fixed an issue where, if several realm deployments are performed, and the FTL API attempts to connect to a persistence cluster that is not running, the API call can hang for longer than expected before failing.
FTL-14161	Fixed an issue where, in rare cases, the FTL server could crash on shutdown.
FTL-14138	Fixed a small memory leak that could occur on reconnect to FTL server.
FTL-13934	When an extremely large disk persistence file is compacted, the persistence service may experience a brief stall.
FTL-13924	When disk persistence is enabled, messages in replicated stores are written to the persistence service's data directory. If non-replicated messages are swapped to the same disk, the persistence service could exceed max.disk.fraction when writing.
FTL-13923	Fixed an issue where the "max.disk.fraction" feature did not take into account messages swapped from non-replicated stores (or messages swapped from replicated stores when disk persistence is not enabled).
FTL-13419	Fixed an issue that could cause the persistence service to exit abruptly when purging a map with async disk persistence enabled.
FTL-12709	Fixed an issue where, at very high message rates, a route might not get re-established following a network or quorum disruption.
FTL-12435	Fixed an issue related to store forwarding where, if the remote cluster has async disk persistence and suffers data loss (e.g., due to multiple disk failure), the local cluster might lose messages sent after the data loss event.
FTL-11185	If the cluster message swapping setting disk_mode is set to swap and the disk is

Key	Summary
	being accessed via NFS, client or quorum timeouts of up to a few seconds can occur.

Known Issues

The following issues exist in this release of TIBCO FTL® - Enterprise Edition:

Key	Summary
FTL-15034	When a FTL Server is setup with user defined certs and If a user is logged in to a FTLServer UI from a browser window, subsequently, if the user attempts to open a UI session to another FTL Server from the same cluster on the same host or to another FTL Server that's behind a load balancer, then FTL Server logs out the first user session and presents a login page for the new FTL Server UI session.
FTL-13960	**Summary:** Sending extremely large messages, for example 1 GB messages, could disrupt the FTL persistence quorum. **Workaround:** Increase the `pserver_timeout_pserver` for the affected persistence cluster.
FTL-13790	**Summary**: After upgrading one FTL server to 6.9.0 or later, another FTL server at 6.8.0 or lower may log a message like the following: Error processing cluster message: invalid type. **Workaround**: None. There is no functional impact. The warning may be ignored.
FTL-13411	**Summary**: On macOS, when the installation package is downloaded through a web browser, it may get labeled as quarantined by the operating system. Installation may result in a system prompt stating that the package cannot be opened. **Workaround**: Remove the quarantine flag from the package before installing it. For example: 6.9.0 and later: `xattr -d com.apple.quarantine TIB_ftl_7.0.1_macos_x86_64.pkg` Before 6.9.0: `xattr -d com.apple.quarantine TIB_ftl_6.8.0_macosx_x86_64.pkg`
FTL-13171	**Summary**: The `use_endpoint_store_for_inbox` and `enable_permissions`

Key	Summary
	features require 6.8.0 or later clients. If an older client is sending or receiving inbox traffic on an endpoint with no direct path transports, and either feature is enabled, the older client should raise an exception during the realm deployment. The older client either accepts the deployment or replies with needs restart. In both cases, the older client is not functional and must be upgraded to 6.8.0 or later. **Workaround**: Upgrade clients to 6.8.0 or later when using the `use_endpoint_store_for_inbox` and `enable_permissions` features.
FTL-13170	**Summary**: The realm property `use_endpoint_store_for_inbox` requires 6.8.0 or later clients. If this property is set to true, and a client older than 6.8.0 attempts an inbox send on an endpoint with no direct-path transports, the old client crashes rather than raising an appropriate exception. **Workaround**: Upgrade clients to 6.8.0 or later when using the `use_endpoint_store_for_inbox` feature.

FTL-12794	**Summary**: In the Go API in FTL 6.8.0, a submessage can be set in with two different value types:
	1. `ftl.Message`
	2. `MsgContent map[string]interface`{}
	A `MsgContent` is returned by:
	`func (m Message) Get (fields ...string) (c MsgContent, err error)`
	If the message is a submessage field the `c MsgContent` has another `MsgContent` for the corresponding submessage.
	When the second method is used to set a submessage and the application is configured to manage all formats, then the APIthrows an exception like the following:
	Exception:
	`TIBEX_SET_ERROR(e, TIB_NOT_PERMITTED, "Format is not defined for "`
	`"this application and all formats must be managed "`

Key	Summary

	`"- cannot create a dynamic format.");`
	
	Workaround:
	Assumptions:
	1. The configuration is set to `manage_formats = true` or `manage_all_formats = true`.
	2. The Go client created a message with a statically configured format that has a
	submessage:
	`msg, _ := realm.NewMessage("outermessage")`
	`subMsg, _:= realm.NewMessage("innermessage")`
	
	Procedure:
	To create the correct formats for the message and submessage (example):
	1. Set some fields:
	`subContent, _ := subMsg.Get()`
	`subContent["someLongField"] = 123455678940000`
	`content, _ := msg.Get()`
	`content["longname-field"] = 256`
	`content["submessage-field"] = subMsg`
	`content[FieldNameString] = "internal msg"`
	`msg.Set(content)`
	2. Add one more field as follows:
	`allcontent, _ := msg.Get()`
	`allcontent["newFieldString"] = "newly added field value"`
	`msg.Set(allcontent)`
	
	We have created the correct formats for the message and submessage.
	The last `msg.Set()` operation fails even though the correct formats are created.

FTL-12619	**Summary**: FTL 6.4 clients that have endpoints configured with server-based transports cannot use request/reply calls or request/reply inboxes with FTL 6.5 or later clients. **Workaround**: Upgrade the clients to 6.5 or later.
FTL-12580	**Summary**: Although the API to backup a realm database (`api/v1/server`

Key	Summary
	`command backup`) returns 200 on success, the backup has not necessarily completed when the request returns. The response body should contain a status code of 202. The status code of the HTTP response itself is 202, which correctly indicates the semantics of the backup request, namely that the backup may not have been completed when the response is returned. **Workaround**: None.

FTL-12517	**Summary**: On Linux Platforms, if you have both FTL 6.7.0 and eFTL 6.7.0 installed and are using yum or zypper package managers to upgrade to FTL/eFTL 6.7.1, the upgrade procedure can fail.
	Workaround: When installing, follow these steps:
	1. Install FTL/eFTL 6.7.1 together
	For yum:
	`yum  install -y TIB_ftl_6.7.1/rpm/*.rpm TIB_eftl_6.7.1/rpm/*.rpm`
	For zypper:
	`zypper install TIB_ftl_6.7.1/rpm/*.rpm TIB_eftl_6.7.1/rpm/*.rpm`

FTL-12181	**Summary**: For the Administrative GUI, resizing the window sometimes hides the vertical scrollbar. **Workaround**: Resize to a larger window or avoid resizing.
FTL-11597	**Summary**: If an FTL client creates many subscribers on durables using async acks and a low ack batch time, the client experiences high CPU usage. **Workaround**: None.
FTL-11185	**Summary**: If the cluster message swapping setting `disk_mode` is set to swap and the disk is being accessed via NFS, client or quorum timeouts of up to a few seconds can occur.

Key	Summary
	Workaround: Use a local filesystem for the location of swap files, via the new cluster message swapping setting `swapdir`.
FTL-10631	**Summary**: Inbox subscribers do not explicitly acknowledge message delivery if the subscriber's endpoint has no direct path transports associated. *Note:* This applies to FTL releases prior to 6.8.0 or if `use_endpoint_store_for_inbox` is set to false. **Workaround**: None.
FTL-10393	**Summary:** FTL 6.3.x or later versions are not compatible with EMS 8.5.0 or earlier releases. **Workaround**: Upgrade to EMS 8.5.1.
FTL-10335	**Summary**: When 6.2 or later clients connect to a 5.4 realm server, the realm server logs a panic. The 5.4 realm server does not crash, and is functional after it logs a panic. 6.2 clients are able to successfully connect. **Workaround**: Upgrade all FTL servers before upgrading clients.

FTL-9499	**Summary**: Importing realm definition data into Release 6.0.1 that had been output in JSON format from Release 6.0.0 could cause the FTL server to reject the deployment even though it did not report validation errors.
	Workaround: From `_GroupServer` application, remove the definitions of:
	`_clientEndpoint`
	`_inboxEndpoint`
	`_loggingEndpoint`
	`_monitoringEndpoint`

FTL-9293	**Summary**: During migration from Release 5.4 to Release 6.x, it is possible that the old realm server and the new FTL server could both assign ordinals to group members. This could result in thrashing behavior by the group members. **Workaround**: Immediately stop the old 5.4 realm server.
FTL-9281	**Summary**: The REST command to compact an FTL server database is not functional.

Key	Summary
	Workaround: None.
FTL-9231	**Summary**: When FTL servers are not in a quorum, the GUI displays incorrect monitoring data. **Workaround**: None
FTL-8319	**Summary**: On Windows platforms, after silent installation of one installation type, subsequently installing with a different installation is ineffective. **Workaround**: Completely uninstall the previous installation type, and reinstall with a new installation type.
FTL-8280	**Summary**: Subscribers on the monitoring endpoint could miss the final monitoring metrics from a closing client. **Workaround**: None.
FTL-7718	**Summary**: Debian Linux Uninstall After uninstalling Debian Linux packages, the command `dpkg -query` reports that the package `tibco-ftl-thirdparty` is still installed (even though it has, in fact, been successfully uninstalled). **Workaround**: This command resolves this issue by removing package information from the database. `sudo dpkg --purge tibco-ftl-thirdparty`
FTL-7161	**Summary**: The realm server GUI does not support Internet Explorer (IE) 11 and earlier. However, it does support Edge (Windows 10). **Workaround**: Use any supported browser as listed in the file: readme.txt.
FTL-5630	**Summary**: Changing a Persistence Cluster If you change the name of a persistence cluster, all running persistence servers in the cluster require restart. However, the realm server GUI does not detect this condition. **Workaround**: Ensure that you restart persistence servers after changing their cluster's name.

Key	Summary
FTL-4386	**Summary**: Chrome and Safari browsers can no longer access TIBCO HTML documentation from a file system, that is, using the `file://protocol`. **Workaround**: Access using a different browser, or access the HTML documentation through the web, that is, using the `http://protocol`.
FTL-496	**Summary**: On Microsoft Windows platforms (only), `tcp` and `shm` transports support at most 60 simultaneous connections. For example, when 60 tcp transports are connected to one listening transport in an application program (for example, a server hub application), the 61st cannot connect, and FTL logs an error. Similarly, when `shm` transports in 60 application processes on the same host computer are connected to the same shared memory segment, the 61st cannot connect, and FTL logs an error. When inline mode is disabled, this limitation of 60 connections applies separately to each transport. However, when inline mode is enabled - by using the property `TIB_EVENTQUEUE_PROPERTY_BOOL_INLINE_MODE` when creating an event queue - then this limitation applies cumulatively across all the transports of all the endpoints (that is, subscribers) associated with that queue. The sum of all the connections to those transports cannot exceed 60. **Workaround**: None.

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. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

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

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.