TIBCO Enterprise Message Service™ Release Notes

Software Release 8.5.1
October 2019

Important Information

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

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

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

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 TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, Two-Second Advantage, TIBCO Cloud Integration, TIBCO Flogo Apps, TIBCO Flogo, TIB, Information Bus, Rendezvous, and TIBCO Rendezvous are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

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

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

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

Contents

About this Product

TIBCO is proud to announce the latest release of TIBCO Enterprise Message Service™ software.

This release is the latest in a long history of TIBCO products that leverage the power of the Information Bus® technology to enable truly event-driven IT environments. To find out more about how TIBCO Enterprise Message Service software and other TIBCO products are powered by TIB® technology, please visit us at www.tibco.com.

TIBCO Enterprise Message Service software lets application programs send and receive messages according to the Java Message Service (JMS) protocol. It also integrates with TIBCO FTL, TIBCO Rendezvous, and TIBCO SmartSockets messaging products.

TIBCO EMS software is part of TIBCO® Messaging.

Product Editions

TIBCO Messaging is available in a community edition and an enterprise edition.

TIBCO Messaging - Community Edition is ideal for getting started with TIBCO Messaging, for implementing application projects (including proof of concept efforts), for testing, and for deploying applications in a production environment. Although the community license limits the number of production clients, you can easily upgrade to the enterprise edition as your use of TIBCO Messaging expands.

The community edition is available free of charge. It is a full installation of the TIBCO Messaging software, with the following limitations and exclusions:

- Users may run up to 100 application instances or 1000 web/mobile instances in a production environment.

- Users do not have access to TIBCO Support, but you can use TIBCO Community as a resource (https://community.tibco.com).

- Available on Red Hat Enterprise Linux Server, Microsoft Windows & Windows Server and Apple macOS.

TIBCO Messaging - Community Edition has the following additional limitations and exclusions:

- Excludes Fault Tolerance of the server.

- Excludes Unshared State Failover.

- Excludes Routing of messages between servers.

- Excludes Central Administration.

- Excludes JSON configuration files.

TIBCO Messaging - Enterprise Edition is ideal for all application development projects, and for deploying and managing applications in an enterprise production environment. It includes all features presented in this documentation set, as well as access to TIBCO Support.

TIBCO Documentation and Support Services

How to Access TIBCO Documentation

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

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit https://docs.tibco.com.

TIBCO Enterprise Message Service Documentation

The following documents for this product can be found on the TIBCO Enterprise Message Service ™ product documentation page:

- TIBCO Enterprise Message Service User's Guide Read this manual to gain an overall understanding of the product, its features, and configuration.

- *TIBCO Enterprise Message Service Central Administration* Read this manual for information on the central administration interface.

- TIBCO Enterprise Message Service Installation Read the relevant sections of this manual before installing this product.

- TIBCO Enterprise Message Service C & COBOL Reference The C API reference is available in HTML and PDF formats.

- TIBCO Enterprise Message Service Java API Reference The Java API reference can be accessed only through the HTML documentation interface.

- *TIBCO Enterprise Message Service .NET API Reference* The .NET API reference can be accessed only through the HTML documentation interface.

- *TIBCO Enterprise Message Service Installation on Red Hat OpenShift Container Platform:* This manual describes how to run TIBCO Enterprise Message Service servers on the Red Hat® OpenShift Container Platform.

- TIBCO Enterprise Message Service Release Notes Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release. This document is available only in PDF format.

Other TIBCO Product Documentation

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

- TIBCO FTL®
- TIBCO Rendezvous®
- TIBCO SmartSockets®
- TIBCO EMS® Client for z/OS (CICS)
- TIBCO EMS® Client for z/OS (MVS)
- TIBCO EMS® Client for IBM i

Third-Party Documentation

- Java™ Message Service specification, available through http://www.oracle.com/technetwork/java/jms/index.html.

- *Java™ Message Service* by Richard Monson-Haefel and David A. Chappell, O'Reilly and Associates, Sebastopol, California, 2001.

- Java™ Authentication and Authorization Service (JAAS) LoginModule Developer's Guide and Reference Guide, available through http://www.oracle.com/technetwork/java/javase/jaas/index.html.

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit http://www.tibco.com/services/support.

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at https://support.tibco.com.

- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to https://support.tibco.com. If you do not have a user name, 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 https://community.tibco.com.

New Features

This section lists features added since the last major (8.0.0) release of TIBCO Enterprise Message Service.

Release 8.5

The following new features have been added to version 8.5 of TIBCO Enterprise Message Service.

.NET Core Support

The EMS .NET client library files, `TIBCO.EMS.dll`, `TIBCO.EMS.ADMIN.dll` and `TIBCO.EMS.UFO.dll` are now built to the .NET Standard 2.0 specification. They can be used to build both .NET Framework applications that can only run on Windows and .NET Core applications that can run on both Windows and Linux.

Platform Requirements for .NET Core:

- Red Hat Enterprise Linux Server
- Microsoft Windows
- Microsoft Windows Server

Tool Requirements:

- To build .NET Core programs: Microsoft Visual Studio 2017 and the associated toolchain or the .NET Core 2.1 SDK (LTS)
- To build .NET Framework programs: Microsoft Visual Studio 2017 and the .NET Framework 4.7.2 Developer Pack or the .NET Core 2.1 SDK (LTS)
- To run .NET Core programs: .NET Core 2.1 (LTS)
- To run .NET Framework programs: .NET Framework 4.7.2 (Windows only)

Package Requirements:

- Programs that involve LDAP JNDI lookups need to reference the `System.DirectoryServices.Protocols` package. This can be done in Visual Studio or in C# project files (`*.csproj`).
- An example is provided in the `EMS_ROOT/samples/cs` folder.

Limitations:

- .NET Core 2.1 does not support distributed transactions.
- LDAP JNDI lookups are supported for .NET Core only on Windows.

New *TIBCO Enterprise Message Service Installation on Red Hat OpenShift Container Platform* book

This release adds a book to the TIBCO Enterprise Message Service documentation, which describes how to install, configure, and run servers in a fault-tolerant configuration using Docker® and the Red Hat OpenShift Container Platform. TIBCO EMS on OpenShift Container Platform is supported for the following versions of the products and components involved:

- TIBCO EMS 8.5.0 and later
- TIBCO FTL 5.4.1 and later (static TCP transports only)

- Docker 18
- Red Hat Enterprise Linux 7.4
- Red Hat OpenShift Container Platform 3.11

TIBCO EMS on OpenShift Container Platform supports all EMS features, with the following exceptions:

- Excludes transports for TIBCO Rendezvous®
- Excludes transports for TIBCO SmartSockets®
- Excludes stores of type `dbstore`
- Excludes stores of type `mstores`

Server Health Check

On Linux, macOS, and Windows, the EMS server can now be configured to expose an HTTP port on which to service health checks. It can then be asked through an HTTP GET request if it is running and, separately, if it is in the active state. This can be used to support the health check probes in an OpenShift cluster.

CRC Calculations Performance Improvement

The EMS server uses CRC to validate data integrity in store files. On Linux, macOS, and Windows, the server now takes advantage of a faster hardware-based way of calculating CRCs on machines equipped with processors that support the `PCLMULQDQ` instruction. On other platforms, it now uses an implementation of the CRC algorithm that is more performant as well.

Line Editing in the Administration Tool

The administration tool now supports line editing and command history on all platforms.

Enhancements to the JAAS Authentication Modules

- The Connection Limit JAAS authentication module can now be configured to apply connection limits to users identified by the combination of their `LDAPID` and hostname through the new `LDAPID@HOSTNAME` user type.
- The timeout of LDAP connect and read operations in the prebuilt JAAS modules is now configurable through the new `tibems.ldap.operation_timeout` property.

Release 8.4

The following new features have been added to version 8.4 of TIBCO Enterprise Message Service.

Filtering Messages in the Undelivered Message Queue

You can now filter messages in the undelivered message queue by destination using a selector. In `$sys.undelivered`, the `JMSDestination` header field can be used in a selector the same way that a supported header field or any other message property with a string value is used.

Flow Control in the Absence of Consumers

The EMS server now enforces flow control on destinations regardless of the presence of consumers. Prior to this release, if there was no message consumer for a destination, the server would not enforce flow control for the destination. For now, this behavior can be restored through a server property. That option will be removed in a future release.

Connection Limit JAAS Authentication Module

This new JAAS authentication module limits the number of active connections a user can have at any one time.

General Enhancements

- Depending on the nature of the error, a server that experiences a file write error either exits or tries writing again. You can now configure the server so that it always exits if it cannot write to a file, using the new `always_exit_on_disk_error` server property with the administration tool, Admin API and Central Administration. That property can be changed dynamically without restarting the server.

- On Windows, this release now supports the **Automatic (Delayed Start)** Windows Service startup type in addition to the previously supported **Manual** and **Automatic** types.

- You can now configure the server so that it requires digital certificates for SSL connections coming from routes but not from clients or its fault-tolerant peer.

- Prior to this release, when the server had successfully sent a message to a consumer through a queue backed by a store file, that message was swapped out to disk. If a consumer then closed its session before handling all the messages it had received, those messages were returned to the queue. If that consumer, or another one, later came back to that queue, those messages would need to be swapped back in from disk. The performance of queues with a high `prefetch` value would be most affected by this swapping pattern. To limit unnecessary swapping, swapping out is now performed, in this context, only when a queue has reached the value set through the `destination_backlog_swapout` server property.

- With a server running under a light load, the messages delivered to a selector-based subset of consumers on a queue could be distributed to that subset unevenly as compared to the messages delivered to the rest of the consumers. With this release, those messages are now distributed more evenly.

- In order to convert into `.json` files legacy `.conf` files that had been written using various encodings, the `tibemsconf2json` utility now takes a parameter to specify the character set used in the original `.conf` files.

- EMS now ships with .NET Admin API sample programs.

- The server could reject a client's reconnection attempts for a while due to a lingering internal reference to the corresponding connection ID. In this particular case, the server will now accept the client's reconnection attempt right away.

Logging and Tracing Enhancements

- A destination is unbounded when it does not have either its `maxbytes` or its `maxmsgs` property set. By default, destinations are unbounded. Unbounded destinations can affect performances and use up the server's available memory if they become backlogged. Starting with this release, the server attempts to detect when such a destination is growing too large, logging a warning when it does. The new `large_destination_count` and `large_destination_memory` properties control the detection thresholds. If not set, the server establishes its own thresholds.

- Attempts to perform an operation by a user who does not have the permission to perform it are now traced in the server log file.

- The logging of low level errors occurring upon writing to a file has been improved.

- The `route disconnected` trace statement has been improved to indicate the reason.

- The client ID is now included in most connection-related trace statements in the server log file.

- The previous release introduced new trace statements for slow operations such as long writes to store files on slow disks. Not all of these statements included the duration of the slow operation. With this release, all slow operation trace statements now report an approximate duration.

- The server now also traces a sync operation on a the `meta.db` store that takes too long.

- Client trace statements for creating a connection now include the active URL being effectively used to establish the connection in addition to the factory (potentially fault-tolerant) URL.

Administration Tool Enhancements

- The `show server` command in the administration tool and on the equivalent page in Central Administration now show separate counts for client connections and admin connections. This extends to the corresponding methods in the Admin API.

- The `show connections full` command in the administration tool and on the equivalent page in Central Administration now show the ephemeral TCP port for each connection. This extends to the corresponding methods in the Admin API. Note that the ephemeral port was already available in the server log when the `trace_client_host` property was configured with `both_with_port`.

Central Administration Enhancements

- The Central Administration server now supports HTTPS.

- The deployment of a configuration fails if a target EMS server has been locally modified through the administration tool or the Admin API. With this release and under those circumstances, you can now view the changes made in the EMS server or those made in Central Administration since the last refresh. You can then choose between refreshing the Central Administration configuration with the EMS server's configuration or overwriting the EMS server configuration with Central Administration's configuration.

- Central Administration can now be installed on its own using the new universal installer Central Administration profile.

- You can now rename an EMS server configuration.

- If the JSON source file of a configuration is too large to render in a reasonable amount of time, you can now either display an unformatted version or download it.

Release 8.3

The following new features have been added to version 8.3 of TIBCO Enterprise Message Service.

TLS v1.2 Support

With the addition of TLS v1.2 cipher suites, this release supports TLS v1.2.

Slow Operation Tracing

The EMS server has been tracing slow operations, such as long writes to store files on slow disks, using the following generic error message: `ERROR: Slow clock tick 15, delayed messaging and timeouts may occur`. This release provides improved diagnostic information in the majority of circumstances. For example:

```
WARNING: A single write to store ($sys.failsafe) lasted around 15 seconds.
```

Compaction and Truncation for mstores

In additions to stores of type `file`, you can now compact stores of type `mstore`. Furthermore, you can now configure the EMS server to occasionally attempt to truncate an mstore, relinquishing unused disk space. For more information, see the sections describing the `compact` command and the `mstore_truncate` property in the *TIBCO Enterprise Message Service User's Guide*.

Creating and Removing File-Based Stores through Central Administration

When you add or delete a file-based store using Central Administration, you no longer need to restart both the active and standby EMS servers in a fault tolerant pair before the changes take effect. You can now restart only the active server, and the standby server picks up the changes when it activates. This means, for fault-tolerant pairs of servers, that restarting only the active server is now enough to make the additions or deletions of stores of type file effective on both sides.

Note that this applies only to changes made through Central Administration, and only to stores of type `file` (not to those of type `mstore` nor `dbstore`). Only additions or deletions of stores are picked up; if you modify an existing store, you must still restart both servers.

EMS Server Backup Files

The EMS server now keeps a copy of the previous JSON configuration file upon receiving a deployment from the Central Administration server. When the deployment is successful, the previous configuration is kept in a file of the same name as the current configuration with an additional `.bak` suffix. For example, `tibemsd.json.bak`. During subsequent deployments, the backup file is overwritten.

General Enhancements

- You can now configure the interval at which an EMS server attempts to connect or reconnect a route to another server using the new `active_route_connect_time` server property.

- The new `JNDI` trace option allows you to print a trace message for each JNDI lookup performed by a client, including the name and type of the object looked up and its return value.

- BSD sockets are now supported in the EMS C client on HP-UX platforms.

- The value of the `module_path` server property is now available through the Administration Tool, Admin API and Central Administration.

- The previous release introduced improvements for queue consumers with a selector where the queue has a large message backlog. One of these improvements involved caching message properties in EMS server memory. It only applied to new incoming messages whereas messages already existing in the backlog were not optimized. With this release, messages already existing in the backlog are now also optimized.

- Other minor improvements were made to the performance of the EMS server.

- When fault tolerance is configured to use SSL in between servers, both servers may use distinct certificates, which may hold different values for their CN fields. This has been supported for classic configurations files and is now also supported with JSON configuration files.

- The `show transport` Administration Tool command and its Admin API counterpart now return all of the properties of an FTL transport, even if the initialization of that transport failed.

- The EMS server now reports a warning message when a client fails to create a producer on a temporary destination that includes the server name as its second member. For example, a destination that starts with `$TMP$.`*server-name*. Previously, warnings were only issued following a failure to create a consumer under the same circumstances.

- The message traced by the EMS server in case of LDAP authentication failure has been improved.

Central Administration Enhancements

A variety of enhancements were made to the Central Administration feature, including the following.

- **Running the EMS Central Administration Server as a Windows Service**

  You can now register the Central Administration server to run as a Windows Service, allowing it to be started automatically. For details, see the *TIBCO Enterprise Message Service Central Administration* guide.

- **Monitoring Destinations with Central Administration**

  When using Central Administration to monitor destination activity on an EMS server, you can now retrieve records for queues and topics by page, with no limits placed on the number of pages that you can request. Note that the resulting records are unsorted.

  To enable this feature, a Retrieve By Page option has been added to the Retrieve Count tool in the monitoring results page for queues and topics.

- **Additional Validation**

  - When the User Authentication Source list includes `ldap`, Central Administration now verifies that the required LDAP properties are provided through the LDAP page.

  - When Authorization is enabled and Fault Tolerance configured, Central Administration now verifies that a user with the same name as the Server Name is present in the configuration.

Release 8.2

The following new features have been added to version 8.2 of TIBCO Enterprise Message Service.

EMS Transport for TIBCO FTL

TIBCO Enterprise Message Service can now exchange messages with TIBCO FTL using the EMS transport for TIBCO FTL. This feature is supported on those platforms where TIBCO FTL is supported. Please refer to the respective readme files of TIBCO Enterprise Message Service and TIBCO FTL.

For more information, see the section *Interoperation with TIBCO FTL* in the *TIBCO Enterprise Message Service User's Guide*.

Temporary Destination Properties

This release introduces support for setting the properties `maxBytes`, `maxMsgs` and `overflowPolicy` on temporary topics and queues through the use of the temporary destination wildcard `$TMP$.>`.

For more information, see the section on Temporary Destination Properties in the *TIBCO Enterprise Message Service User's Guide*.

Queue Selector Improvements

This release introduces improvements for queue consumers with a selector where the queue has a large message backlog.

One improvement enhances the performance of such queue consumers by caching message properties in EMS server memory. As a result, you may see an increase of the memory footprint of the server if you have such queue consumers. If needed, please adjust the `max_msg_memory` server property, which controls how much memory the server uses for messages.

Note that for a given queue, the optimization is triggered the first time a consumer with a selector is created. However, only new incoming messages are optimized; messages already existing in the backlog are not optimized through the server cache. If the server is restarted and a fault tolerant consumer on the queue is restored, then all recovered messages in that queue are optimized.

Another improvement enhances the responsiveness of the EMS server in the same situation. This was achieved by implementing a time slicing mechanism.

Asynchronous Write Mode for mstore

The store type `mstore` now supports the asynchronous write mode. For details, see the `mode` parameter description in the `stores.conf` section of the *TIBCO Enterprise Message Service User's Guide*.

Additionally, this release introduces mstore performance improvements.

Enhancements to the .NET API Reference

The generated reference documentation for the .NET API has been updated with a new and improved look and feel. The .NET API reference can be accessed through the HTML documentation interface.

Originating Connections for Temporary Topics

This release includes the new `DestinationInfo.connectionID()` admin API Java method and its .NET equivalent that link a temporary topic back to its connection of origin. For more information, see the Java and .NET API Reference documentation, accessible through the HTML documentation interface.

Converting a Pair of Fault Tolerant Servers to JSON

The `tibemsconf2json` utility can now convert the text-based EMS server configuration files of both servers in a fault tolerant pair to a single JSON configuration file. For details, see the appendix on Converting Server Configuration Files to JSON in the *TIBCO Enterprise Message Service Central Administration* guide.

Timeout for Outgoing Route Connections

The `handshake_timeout` parameter for the EMS server, in addition to controlling the wait time for an incoming SSL connection to complete, now also controls the amount of time that the EMS server waits for an outgoing route connection (TCP or SSL) to complete.

Release 8.1

The following are new features in this release:

JAAS Authentication Modules

TIBCO now supports several compiled and fully functional JAAS modules that can be used to authenticate users in the EMS server. For more information, see *JAAS Authentication Modules* in the *TIBCO Enterprise Message Service User's Guide*.

Topic Prefetch Property for Routes

You can now specify a prefetch value for topics at the route level. This allows you to assign larger values for WAN routing functions.

If `topic_prefetch` is not set, the route uses the `prefetch` value specified for the topic. If a `topic_prefetch` is set for the route and a different `prefetch` is set for the topic, the `topic_prefetch` value overrides the destination `prefetch`.

Both properties are described in the *TIBCO Enterprise Message Service User's Guide*.

Secondary Log Files

JSON-configured servers in fault-tolerant mode can now specify separate log files for the primary and secondary servers in the pair. See the `secondary_logfile` parameter description in the *TIBCO Enterprise Message Service User's Guide*.

Increased Network Threads

You can now control the number of network threads used by the EMS server without assigning them to specific cores. For more information, see the description for the `network_thread_count` parameter and the section on Increasing Network Threads without Setting Thread Affinity in the *TIBCO Enterprise Message Service User's Guide*.

Documentation Separated from the Product Installer

TIBCO Enterprise Message Service documentation is no longer bundled with the installer. You can obtain the documentation from https://docs.tibco.com/products/tibco-enterprise-message-service.

This link opens documentation for the most recently released version of Enterprise Message Service. Click the version tabs to access documentation for other releases of the product.

Central Administration Features

See the *TIBCO Enterprise Message Service Central Administration* guide for details on these new features.

- **Central Administration Groups**

  You can now change the default JAAS groups that are used to authenticate users when JAAS is enabled.

  - `--jaas-admins` allows you to change the groups given administrative privileges in Central Administration.

  - `--jaas-guests` allows you to change the groups given guest privileges in Central Administration.

    For more information on JAAS groups, see the section *Configuration of JAAS Authentication* in the *TIBCO Enterprise Message Service Central Administration* guide.

- **Central Administration Max Deployments**

  The `--keep-max-deployments` option for Central Administration allows you to limit the number of deployments kept in the Recent Deployments list.

.NET Library Support for JMS 2.0

For information on the .NET library and functions, see the see the .NET API reference, accessible through the HTML documentation interface.

C Library Support for JMS 2.0

See the *TIBCO Enterprise Message Service C & COBOL Reference* for more information on these functions.

- **Shared Subscriptions**

  The following new functions are added to the Shared Subscriptions feature:

  - `tibemsSession_CreateSharedConsumer`

  - `tibemsSession_CreateSharedDurableConsumer`

  - `tibemsAdmin_GetSubscriptions`

  - `tibemsSubscriptionInfo`

- **Asynchronous Sending**

  The C API now supports the asynchronous sending feature, which permits message producers to send messages asynchronously, offloading the notification of the success or failure to another thread.

  - `tibemsMsgProducer_AsyncSend`

  - `tibemsMsgProducer_AsyncSendEx`

  - `tibemsMsgProducer_AsyncSendToDestination`

  - `tibemsMsgProducer_AsyncSendToDestinationEx`

    — `tibemsMsgCompletionCallback`

- **Delivery Delay**

  The C API now supports the Delivery Delay feature, which permits message publisher to specify a delivery time for messages. The EMS server will only deliver the message after the delivery time specified when the message is published.

  — `tibemsMsgProducer_SetDeliveryDelay`

  — `tibemsMsgProducer_GetDeliveryDelay`

  — `tibemsMsg_GetDeliveryTime`

Release 8.0

The following are new features in this release:

Support for JMS 2.0

This release adds support for the JMS 2.0 specification. Currently, this support is offered only to Java clients. The features added with JMS 2.0 include:

- **Delivery Delay**

  Message publishers can now specify a delivery time for messages. The EMS server will only deliver the message after the time delivery time specified when the message is published. For more information, see the section on Delivery Delay in the *TIBCO Enterprise Message Service User's Guide*.

- **Asynchronous Sending**

  Message producers can now send messages asynchronously, offloading the notification of the success or failure to another thread and thereby increasing performance in certain situations. For details, see the section on Sending Messages Synchronously and Asynchronously in the *TIBCO Enterprise Message Service User's Guide*.

- **Shared Subscriptions**

  An application can now share the work of message consumption across multiple topic consumers. When message consumers share a subscription to a topic, only one consumer will receive a published message. For details, see the section on Shared Subscriptions for Topics in the *TIBCO Enterprise Message Service User's Guide*.

  Additionally, the following new Java admin API methods implement the Shared Subscriptions feature:

  — `ConsumerInfo.getSharedSubscriptionName`

  — `ConsumerInfo.isShared`

  — `TopicInfo.getDurableSubscriptionCount`

  — `TopicInfo.getSubscriptionCount`

      For details on these Java admin API methods, see the *API Reference*, available through the HTML documentation. For details on the equivalent C and .NET admin methods, see the *TIBCO Enterprise Message Service C & COBOL Reference* and the *API Reference*.

- **Simplified API**

  In addition to the API provided with the JMS 1.1 specification, which is now called the Classic API, the JMS 2.0 specification offers a simpler and less verbose API called the Simplified API. For details, see the section on the JMS 2.0 Specification in the *TIBCO Enterprise Message Service User's Guide*.

Central Administration Monitoring

The monitoring feature allows you to see various metrics (depending on level of statistics configured in the EMS server) as well as runtime configuration settings. For details, see the *Monitor the Servers* section in the *TIBCO Enterprise Message Service Central Administration*.

IBM System SSL Application ID

The new function `tibemsSSLParams_System_SetApplicationId` sets the application ID for IBM System SSL implementations.

For more information, see the *TIBCO Enterprise Message Service C & COBOL Reference*.

Logging Enhancement

A new `tibemsd` parameter has been introduced that allows you to specify the maximum number of log files you want to keep. See the description for `logfile_max_count` in the *TIBCO Enterprise Message Service User's Guide*.

JNDI Lookup

The EMS .NET API now supports JNDI lookup of `EMSDTCConnectionFactory` objects. An administrator can now create a `EMSDTCConnectionFactory` in JNDI and an EMS .NET application will be able to look it up using either `LookupContext` or `LdapLookupContext`

Changes in Functionality

This section lists changes in functionality since the last major release of TIBCO Enterprise Message Service.

Release 8.5

The following are changes in functionality in version 8.5 of TIBCO Enterprise Message Service.

Supported Databases

TIBCO Enterprise Message Service 8.5.1 supports the database store feature with Oracle version 12.2 and Oracle Real Application Clusters (RAC) version 12.2.

Native Installers

EMS can now be installed with platform-native installers on Linux, macOS, and Windows. Simple archive options are also available for some of these platforms. TIBCO Universal Installer packages are no longer available for these platforms.

Location of the .NET Client in the Windows GAC

Along with the other APIs, the .NET version of the EMS client is now 64-bit only. Its location in the Windows GAC has changed accordingly.

OpenSSL

TIBCO Enterprise Message Service 8.5.0 operates with OpenSSL version 1.0.2r.

TIBCO Enterprise Message Service 8.5.1 operates with OpenSSL version 1.1.1c.

The cipher suites that are supported by TIBCO Enterprise Message Service 8.5.1 have been revised. For a current list of supported cipher suites, run the `help ciphers` command in the `tibemsadmin` tool. Note that this list is only relevant to the release of TIBCO Enterprise Message Service that ships with the particular version of `tibemsadmin` that is running when the command is issued.

OpenLDAP

TIBCO Enterprise Message Service 8.5.1 operates with OpenLDAP version 2.4.47.

Microsoft Visual Studio 2017

TIBCO Enterprise Message Service is now designed for use with Microsoft Visual Studio 2017. Visual Studio 2015 (including VC14) is no longer supported. C developers on Windows platforms must upgrade to Visual Studio 2017 (including VC15). .NET Framework developers must upgrade to that version as well or switch to the .NET Core SDK.

Release 8.4

The following are changes in functionality in version 8.4 of TIBCO Enterprise Message Service.

Microsoft .NET Framework Redistributable Package

On Windows, version 8.4.1 of TIBCO Enterprise Message Service no longer ships with the Microsoft .NET Framework redistributable package.

If you run EMS .NET client applications, you now need to download and install .NET Framework 4.6.1. We strongly recommend that you apply the latest updates made available by Microsoft through Windows Update for that version.

Flow Control in the Absence of Consumers

With this release, the server enforces flow control on destinations regardless of the presence of consumers.

Prior to this release, if there was no message consumer for a destination, the server would not enforce flow control for the destination. That is, if a queue had no started receiver, the server did not enforce flow control for that queue. Also, if a topic had inactive durable subscriptions or no current subscriber, the server did not enforce flow control for that topic. For topics, if flow control was set on a specific topic (for example, `foo.bar`), flow control was enforced as long as there were subscribers to that topic or any parent topic. For example, if there were subscribers to `foo.bar` or to parent topic `foo.*`, flow control would be enforced on `foo.bar`.

This behavior can be restored by setting the `flow_control_only_with_active_consumer` property, but note that this property and the corresponding behavior are deprecated and will be removed in a future release.

Java Client SSL Support

Previous releases of TIBCO Enterprise Message Service have relied on TIBCrypt, a library provided by TIBCO to perform low level operations to support SSL on Java clients. These were functionalities that did not exist in earlier implementations of the Java Runtime Environment (JRE). Now that those functionalities are available in the latest Java distributions, it is no longer necessary for TIBCO to provide them through a private library.

This release no longer includes the TIBCrypt library in the EMS distribution, and the EMS Java client library now uses the native Java methods. For normal Java client applications that use SSL, this change will be transparent to the user. There are however two notable exceptions:

- Error scenarios in which an exception is thrown. Those exceptions that had been produced within the TIBCrypt library will now be generated by the native Java implementation and will have a different text associated with the exception. Any application that is expecting specific text to appear in an exception will possibly need to be modified (in general, expecting a specific text in an exception should be avoided).

- Support for the Federal Information Processing Standard (FIPS), Publication 140-2 Compliant Java clients. In previous releases, the TIBCrypt library provided an interface to the Entrust implementation of SSL, which could be used instead of the standard JSSE. The Entrust implementation of SSL could operate in a FIPS 140-2 compliant mode. Support for the Entrust implementation had been deprecated in a previous release and has been removed in this release.

  To provide support for FIPS 140-2 compliant Java clients, you may choose to use a FIPS 140-2 compliant cryptographic token provider that supports the PKCS #11 interface (see the "PKCS#11 Reference Guide" -- https://docs.oracle.com/javase/8/docs/technotes/guides/security/p11guide.html). You are responsible for procuring the cryptographic token (hardware or software). You or the token provider should configure the token.

  The old `com.tibco.security.FIPS` TIBCrypt property has been replaced with a new `com.tibco.tibjms.ssl.PKCS11` property (see Javadoc). Existing Java applications that set the old property (to any value) will stop working and should implement a PKCS#11 solution instead

OpenSSL

TIBCO Enterprise Message Service 8.4.0 and 8.4.1 operate with OpenSSL version 1.0.2k.

FIPS Compliance

Support of FIPS Compliance on Solaris (SPARC) platforms, which had been dropped since release 8.2, has been reinstated.

The server supports FIPS compliance only on Windows, Linux, and Solaris (x86 and SPARC) platforms.

Microsoft Visual Studio 2015

TIBCO Enterprise Message Service is now designed for use with Microsoft Visual Studio 2015. Visual Studio 2010 (including VC10) is no longer supported. C and .NET developers on Windows platforms must upgrade to Visual Studio 2015 (including VC14).

Installing on 32-Bit Linux

On most platforms, the TIBCO Universal Installer bundles a JVM that it uses for its own purposes. On Linux, with this release, the installer bundles a 64-bit JVM that will not run on a 32-bit version of Linux. Should you need to install the EMS client libraries on 32-bit Linux, refer to the corresponding instructions in the Installation manual.

General Changes

- With this release, the server dynamically loads the LDAP libraries that are included in the EMS package.

- With this release, the administration tool dynamically loads the SSL and compression libraries that are included in the EMS package.

- The effect of the `handshake_timeout` property has been clarified and enhanced. Check the corresponding entry in the User's Guide for details.

- Increased the maximum size of a route URL from 160 to 1024 characters.

- Since release 8.1, most commands in the administration tool and most methods in the Java and .NET Admin APIs have been unavailable when a server using a JSON configuration file is not in the active state. In such a situation, the only commands available were `show state`, `shutdown` and `rotatelog`. The `show connections` command has been added to the list of available commands.

- A routed server could receive protocol messages regarding interest in a global topic. If that server did not have a global topic with the same name, an alert stating that this interest was ignored was logged to the `$sys.monitor.route.error` monitoring topic. This alert will now appear in `$sys.monitor.route.warning`.

Release 8.3

The following are changes in functionality in version 8.3 of TIBCO Enterprise Message Service.

OpenSSL

TIBCO Enterprise Message Service 8.3.0 operates with OpenSSL version 1.0.2f.

The cipher suites that are supported by TIBCO Enterprise Message Service have been revised. For a current list of supported suites, see the section on "Supported Cipher Suites" in the *TIBCO Enterprise Message Service User's Guide*.

Data Type Mapping with SmartSockets

In previous releases, the SmartSockets `T_MSG_FT_CHAR` type was converted to an EMS `Byte` type while the EMS `Character` type was converted to the SmartSockets `T_MSG_FT_INT2` type.

With EMS 8.3, the latter doesn't change but a SmartSockets `T_MSG_FT_CHAR` type is now converted to an EMS `Character` type.

EMS Server Backup Files

The EMS server now keeps a copy of the previous JSON configuration file upon receiving a deployment from the Central Administration server. The backup file uses a file of the same name as the current configuration with an additional `.bak` suffix.

In case you already have created a file that holds the name of an EMS server's JSON configuration file with an additional `.bak` suffix (such as `tibemsd.json.bak`), please note that this file will be overwritten the next time you deploy the configuration through Central Administration. If you need to preserve the content of that file, you should give it a different name before upgrading to EMS 8.3.0.

General Changes

- Increased the maximum size of a user or group name from 127 to 255 characters.

- In previous releases, route recovery attempts between two servers would affect the values of the `JMSXDeliveryCount` property and `JMSRedelivered` header field of messages in transit on the routes involved. Starting with this release, such a situation will not affect `JMSXDeliveryCount` and `JMSRedelivered` anymore.

- The "Slow clock tick" log message is no longer labeled as an ERROR. It now appears as a WARNING and cannot be suppressed.

Release 8.2

The following are changes in functionality in version 8.2 of TIBCO Enterprise Message Service.

Rendezvous Libraries

Rendezvous libraries are no longer included in the EMS package.

Users who have enabled Rendezvous transports to exchange messages with EMS must configure the `tibemsd` parameter `module_path` to point to previously installed Rendezvous libraries. That is, if `tibrv_transports=enabled`, then the `module_path` parameter must include a path to the appropriate Rendezvous libraries.

OpenSSL

TIBCO Enterprise Message Service 8.2.0 operates with OpenSSL version 0.9.8zc.

TIBCO Enterprise Message Service 8.2.1 operates with OpenSSL version 0.9.8zd.

TIBCO Enterprise Message Service 8.2.2 operates with OpenSSL version 1.0.1p.

FIPS Compliance

FIPS Compliance is no longer supported on Solaris SPARC platforms.

The EMS server supports FIPS compliance only on Windows, Linux, and Solaris 10 (x86) platforms. On UNIX, only tibemsd64, the 64-bit version of the server, is supported. No 32-bit support is provided.

EMS Libraries

On UNIX systems, a number of libraries located in the *EMS_HOME*/`lib` directory appear in the form of versioned files and unversioned symlinks to those files.

Starting with this release, the EMS server loads the unversioned symlink and validates that the corresponding versioned file has the correct version. If it does not, the server prints out a warning.

Release 8.1

The following are changes in functionality in this release.

- **Rendezvous Libraries Dynamically Loaded**

  With this release of TIBCO Enterprise Message Service, the EMS server dynamically loads Rendezvous libraries that are included in the EMS package.

In the next release, Rendezvous libraries will not be included in the EMS package. Instead, users who have enabled Rendezvous transports to exchange messages with EMS must configure the `tibemsd` parameter `module_path` to point to previously installed Rendezvous libraries. That is, if `tibrv_transports=enabled`, then the `module_path` parameter must include a path to the appropriate Rendezvous libraries.

For software release 8.1.0, the path specified in the `module_path` parameter must point to a Rendezvous release 8.2.0 and later.

- **Administration Tool Commands and Admin API**

  As of this release, most commands in the administration tool and most methods in the Java and .NET Admin API are now unavailable when a server using a JSON configuration file is not in the active state. In such a situation, the only commands and corresponding methods available are `show state`, `shutdown` and `rotatelog`.

- **Microsoft Visual Studio 2010**

  TIBCO Enterprise Message Service is now designed for use with Microsoft Visual Studio 2010. Visual Studio 2005 (also known as VC8) is no longer supported. C and .NET developers on Windows platforms must upgrade to Visual Studio 2010.

Release 8.0

The following are changes in functionality in this release.

- **Hibernate Installation Procedure**

  You can now elect to download and install Hibernate Core for Java during the installation of TIBCO Enterprise Message Service. See the *TIBCO Enterprise Message Service Installation* guide for more information.

- **Installation Options**

  Three Installation profiles are now available, allowing you to choose just the client or server, or to select a full development installation.

- **Administration Tool Commands and Topic Consumers**

  With this release and the introduction of shared subscriptions, the relationship between topic subscriptions and topic consumers has changed. Most importantly, the number of subscriptions to a topic is not always equal to the number of consumers.

  As a result, the output produced by some administration tool commands has changed:

  - `show topics` — now reports the number of subscriptions and durable subscriptions, not the number of consumers.

  - `show topic` — reports the number of subscriptions, durable subscriptions, and consumers. The number of consumers represents the number of *active* (that is non-closed) consumer objects created by applications. Offline or closed durable consumers are not included in the count.

  - `show consumers` and `show stat consumers` — no longer report offline durable subscribers.

    Refer to the *TIBCO Enterprise Message Service User's Guide* for details on these commands.

Deprecated and Removed Features

The following tables list any features that have been deprecated or removed for version 8.5 of TIBCO Enterprise Message Service.

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.

Deprecated Features

Affected Component	Description	Deprecated in Release
Client Libraries	This release deprecates the support of static C client libraries. Support for these libraries will be removed in a future release.	8.5.1
Database Stores	This release deprecates the support of the IBM Db2 and MySQL databases when using stores of type `dbstore`. Support for these databases will be removed in a future release.	8.5.0
LDAP Authentication	User authentication through LDAP can be implemented either by setting the EMS server properties starting in `ldap_` or by using the LDAP JAAS authentication modules. This release deprecates the support of the former feature that relies on `ldap_` properties. Support for this feature will be removed in a future release. We recommend using the newer LDAP JAAS authentication modules instead, which remain fully supported.	8.5.0
LDAP JNDI Lookups in C	This release deprecates the support by the EMS C client of JNDI Lookups in an LDAP server. Support for this feature will be removed in a future release.	8.5.0
SmartSockets Transports	This release deprecates the support of the SmartSockets transports. Support for this feature will be removed in a future release.	8.5.0
TIBCO® EMS Transport Channel for WCF	This release deprecates the support of the TIBCO® EMS Transport Channel for WCF component. Support for this component will be removed in a future release.	8.5.0
Flow Control	The `flow_control_only_with_active_consumer` property to revert to the pre-8.4 behavior has been deprecated and will be removed in a future release.	8.4.0
JAAS and JACI	This release deprecates the `jaas_classpath` and `jaci_classpath` parameters. Users should migrate to the new `security_classpath` parameter.	8.1.0

Affected Component	Description	Deprecated in Release
Client Libraries	The `TopicInfo.getDurableCount` Java admin method and equivalent C and .NET methods are deprecated. Instead, use `TopicInfo.getDurableSubsriptionCount`, and equivalent methods in C and .NET.	8.0.0

Removed Features

Affected Component	Description	Deprecated in Release	Removed in Release
SSL Communication	By virtue of using OpenSSL 1.1.x, FIPS 140-2 is no longer supported.	N/A	8.5.1
SSL Communication	The TLSv1.0 protocol is no longer supported.	N/A	8.5.1
SSL Communication	On Solaris platforms only, this release no longer supports OpenSSL static libraries. OpenSSL dynamic libraries continue to be supported.	N/A	8.5.1

SSL Communication	The `ssl_rand_egd` and `ft_ssl_rand_egd` properties are no longer supported.
	
	The corresponding C API functions are no longer supported:
	
	`tibemsSSLParams_SetRandData`
	
	`tibemsSSLParams_SetRandFile`
	
	`tibemsSSLParams_SetRandEGD`
	
	The corresponding Java and .NET API constants called `SSL_EGD_PARAM` are no longer supported.
		N/A	8.5.1

32-bit Client Libraries	Support for the 32-bit client libraries has been removed. This applies to all three client APIs.	8.3.0	8.5.0
32-bit server executables	Support of the 32-bit server executables has been removed. EMS now only includes 64-bit server executables. Consequently, the 32-bit Windows package has become a client-only package. On UNIX platforms, the corresponding files have lost their -64 suffix. Convenience soft links using the former file names are provided, such as `tibemsd64` -> `tibemsd`.	8.3.0	8.4.0
Multicast	Support of the multicast feature of EMS based on the `tibemsmcd` (multicast daemon) has been removed.	8.3.0	8.4.0

Affected Component	Description	Deprecated in Release	Removed in Release
SSL Communication	The following cipher suites are no longer supported: • `SSL_RSA_WITH_DES_CBC_SHA` • `SSL_DHE_DSS_WITH_DES_CBC_SHA` • `SSL_DHE_RSA_WITH_DES_CBC_SHA`	8.3.0	8.4.0
Entrust SSL Libraries	Support for Entrust libraries with EMS clients for Java using SSL has been removed. For FIPS compliance in the Java client, refer to the Enabling FIPS Compliance section in the User's Guide.	8.2.0	8.4.0
Admin API	The following Admin API restart methods have been removed: • Java: `TibjmsAdmin.restart()` • .NET: `Admin.Restart()`	8.2.2	8.4.0
JNDI lookups in C Clients on Windows	C programs that perform JNDI lookups, whether they use LDAP or not, cannot be statically linked on Windows any longer. Use dynamic linking instead.	N/A	8.4.0
EMS Clients, Version 5.x and Below	Queue browsing by older EMS clients (versions 5.x and below) is not supported.	N/A	8.3.0
FIPS Compliance	FIPS compliance by the EMS server and C client is no longer supported on 32-bit Windows systems. It still is supported on the 64-bit Windows, 64-bit Linux and 64-bit Solaris platforms.	N/A	8.2.2
SSL Communication	The following cipher suites are no longer supported in the EMS Java client: • `SSL_RSA_EXPORT_WITH_DES_40_CBC_SHA` • `SSL_DHE_DSS_EXPORT_WITH_DES_40_CBC_SHA` • `SSL_DHE_RSA_EXPORT_WITH_DES_40_CBC_SHA` • `SSL_RSA_EXPORT_WITH_RC4_40_MD5` • `SSL_RSA_WITH_RC4_128_MD5` The following cipher suites are no longer supported in the EMS .NET client: • `EXP-RC2-CBC-MD5` • `EXP-RC4-MD5` • `RC4-MD5`	N/A	8.2.2

Affected Component	Description	Deprecated in Release	Removed in Release
Stores	The 32- and 64-bit `tibemsdb5revert` executables are no longer included in the TIBCO Enterprise Message Service packages. This tool was used to revert EMS store files from software release 5.x to a format compatible with 4.x.	N/A	8.2.0
C Client API	Because the C API does not support character conversion, `tibemsBytesMsg_ReadUTF` and `tibemsBytesMsg_WriteUTF` have been removed.	N/A	8.1.0
TIBCO Hawk	The `com.tibco.tibjms.admin.hawk` package is no longer included with TIBCO Enterprise Message Service. In order to use TIBCO Hawk to monitor TIBCO Enterprise Message Service, a minimum of TIBCO Hawk version 4.9 is required. As of Hawk 4.9, the `com.tibco.tibjms.admin.hawk` package is built into the installation. • With Hawk 4.9 (that ships with TRA 5.8.0), the microagent is still embedded in `hawk/4.9/lib/tibjmsadmin.jar`, which may cause a conflict with `ems/7.0/lib/tibjmsadmin.jar` if an application needs to access both the microagent and the EMS Admin API. • With Hawk 5.0, the microagent sits in its own archive: `hawk/5.0/plugin/ems/hawkemshma.jar` There should be no conflict.	N/A	7.0.0

Platform Support

Please note the following changes in platform support.

Platform	Status	As of Release	Notes
Apple macOS 10.11	Removed	8.5.0	This release supports Apple macOS 10.13 and 10.14.
Apple macOS 10.12	Removed	8.5.0	This release supports Apple macOS 10.13 and 10.14.
Microsoft Windows 32-bit platforms	Removed	8.5.0	This release drops support for 32-bit client libraries, which results in dropping the 32-bit Windows package.
Microsoft Windows 7 SP1	Removed	8.5.0	This release supports Microsoft Windows 10.
Microsoft Windows 8	Removed	8.5.0	This release supports Microsoft Windows 10.

Platform	Status	As of Release	Notes
Microsoft Windows Server 2008 SP2	Removed	8.5.0	This release supports Microsoft Windows Server 2016.
Microsoft Windows Server 2012	Removed	8.5.0	This release supports Microsoft Windows Server 2016.
Microsoft Windows Server 2012 R2	Removed	8.5.0	This release supports Microsoft Windows Server 2016.
Novell SUSE Linux Enterprise Server 11.4	Removed	8.5.0	This Release supports Novell SUSE Linux Enterprise Server 12.x, 15.
HP OpenVMS	Deprecated	8.5.0	This release deprecates the support of all versions of HP OpenVMS. This platform will no longer be supported in a future release.
HP-UX	Deprecated	8.5.0	This release deprecates the support of all versions of HP-UX. This platform will no longer be supported in a future release.
IBM AIX	Deprecated	8.5.0	This release deprecates the support of all versions of IBM AIX. This platform will no longer be supported in a future release.
Novell SUSE Linux Enterprise Server IBM System z	Deprecated	8.5.0	This release deprecates the support of all versions of Novell SUSE Linux Enterprise Server IBM on System z. This platform will no longer be supported in a future release. Novell SUSE Linux Enterprise Server on x86-64 is unaffected.
Oracle Solaris	Deprecated	8.5.0	This release deprecates the support of all versions of Oracle Solaris (both on SPARC and x86-64). This platform will no longer be supported in a future release.
Red Hat Enterprise Linux Server System z	Deprecated	8.5.0	This release deprecates the support of all versions of Red Hat Enterprise Linux Server on System z. This platform will no longer be supported in a future release. Red Hat Enterprise Linux Server on x86-64 is unaffected.
Apple macOS 10.10	Removed	8.4.0	This release supports Apple macOS 10.11 and 10.12.
IBM AIX 6.1	Obsolete	8.4.0	This release supports IBM AIX 7.1 and 7.2.
Microsoft Windows 32-bit platforms	Dropped support for 32-bit executables	8.4.0	The 32-bit Windows package has become a client-only package. 64-bit platforms on x86-64 are unaffected.

Platform	Status	As of Release	Notes
Novell SUSE Linux Enterprise Server 11.3 IBM System z	Removed	8.4.0	This release supports Novell SUSE Linux Enterprise Server 12.x IBM System z.
Novell SUSE Linux Enterprise Server 11.4 IBM System z	Removed	8.4.0	This release supports Novell SUSE Linux Enterprise Server 12.x IBM System z.
OpenVMS 8.3 Itanium	Removed	8.4.0	This release supports OpenVMS 8.4 Itanium
Red Hat Enterprise Linux Server 5.x	Removed	8.4.0	This release supports Red Hat Enterprise Linux Server 6.x and 7.x
Red Hat Enterprise Linux Server 5.x	Deprecated	8.3.0	This release deprecates the support of Red Hat Enterprise Linux Server 5.x (both 32-bit on x86 and 64-bit on x86-64). RHEL 5.x will no longer be supported in a future release.
Microsoft Windows 32-bit platforms	Deprecated	8.3.0	This release deprecates the support of all Microsoft Windows 32-bit platforms on x86, including Windows 7 SP1, Windows 8, and Windows Server 2008 SP2. 64-bit platforms on x86-64 are unaffected.
HP-UX 11i v2 (B.11.23) on Itanium	Obsolete	8.3.0	
Mac OS X 10.9	Removed	8.3.0	This release supports Mac OS X 10.10 and 10.11.
Novell SUSE Linux Enterprise 11.3	Removed	8.3.0	This release supports Novell SUSE Linux Enterprise Server 12.
Windows Vista SP2	Removed	8.3.0	
OpenVMS 8.2.x Itanium	Obsolete	8.3.0	This release supports OpenVMS Itanium 8.3.
Microsoft Windows Server 2003	Obsolete	8.2.2	
Mac OS X 10.8	Obsolete	8.2.0	
Novell SUSE Linux Enterprise Server 11.0	Obsolete	8.2.0	This release supports Novell SUSE Linux Enterprise Server 11.3.

Migration and Compatibility

The following are instructions on how to migrate from a previous release to version 8.5.0 of TIBCO Enterprise Message Service.

Order of Upgrade

Upon upgrading EMS software already installed on separate machines to a newer version of EMS, it is recommended to upgrade and restart in the following order:

1. Upgrade and restart the Central Administration server.

2. Upgrade and restart all EMS servers.

3. Upgrade and restart EMS clients.

Compatibility with TIBCO FTL

- TIBCO Enterprise Message Service releases 8.2.2 and later are compatible with TIBCO FTL 4.1.x and later.

- TIBCO Enterprise Message Service release 8.2.1 is compatible with TIBCO FTL 4.1.x only.

When deploying EMS 8.3 and later transports for TIBCO Rendezvous 8.4.4 and earlier, you must configure the `module_path` parameter to include the path to the EMS libraries before the Rendezvous libraries.

See the *TIBCO Enterprise Message Service User's Guide* for more information on the `module_path` parameter.

Migrating from Release 8.X

Solaris

On both Solaris platforms, the -64 suffix present in the C client 64-bit library file names has been removed. Additionally, third-party 64-bit libraries have been moved from `lib/64` into the `lib` directory.

Convenience soft links using the former file names (such as `libtibems64.so -> libtibems.so`) are provided. Other platforms are unaffected.

Windows

Releases 8.4.1 and later of TIBCO Enterprise Message Service no longer ship with the Microsoft .NET Framework redistributable package.

If you run EMS .NET Framework client applications, you now need to download and install .NET Framework 4.7.2. We strongly recommend that you apply the latest updates made available by Microsoft through Windows Update for that version.

For other platforms, there are no migration procedures when migrating from an 8.x release.

Migrating from Release 7.X

Updating the Database Schema

The 8.0 release of TIBCO Enterprise Message Service introduced some enhancements and changes to the database store feature. After installing the new version of EMS, you must run the EMS Schema Export Tool with the `-updateall -export` options to apply these changes to your database store implementation.

For more information, see the section on the EMS Schema Export Tool in the *TIBCO Enterprise Message Service User's Guide*.

Reverting to an Earlier Release from Release 8.2, 8.3, 8.4 or 8.5

When you upgrade from EMS 8.1 (or earlier) to EMS 8.2 (or later), the EMS server automatically upgrades your mstore files to the 8.2 format to improve performance. These changes are incompatible with 8.1 and will prevent an 8.1 server from starting if you try to roll back to an older version of EMS.

Additionally, you may chose to manually upgrade your mstore file to the 8.3 format to make the new time-bound compact and `mstore_truncate` features available.

In both cases, the `tibemsdbconvert` tool, included with EMS, can be used to return your mstores either to the 8.2 or to the 8.1 format. See "Using the tibemsdbconvert Tool" in the *TIBCO Enterprise Message Service User's Guide* for more information.

Closed Issues

The table lists issues closed in the listed version of TIBCO Enterprise Message Service.

Closed in Release	Key	Summary
Issues Closed in Release 8.5.1		
8.5.1	EMS-7743	Fixed a defect where, using a server release of 8.3.0 or earlier, and creating a consumer with a selector that is improperly using the JMSDestination message header field, the server did not start or activate after being upgraded to EMS 8.4.0 or later.
8.5.1	EMS-7742	Fixed a defect where creating a consumer with a selector improperly using the JMSDestination message header field on a destination other than the $sys.undelivered queue would result in a misleading out-of-memory error in the client and the server log. The error will now be reported as using an invalid selector. This defect affected Releases 8.4.0, 8.4.1 and 8.5.0.
8.5.1	EMS-7739	Fixed two typos in the `com.tibco.jms.jmsclient.rv` and `com.tibco.tpcl.javax.jms` OSGi bundles that were preventing them from working properly.
8.5.1	EMS-7698	Fixed a defect where the C client could deliver messages with an erroneously empty body to the application. This could occur when the application ran out of memory when receiving messages with very large bodies. In such a situation, an out-of-memory condition will now result in the C client terminating the corresponding connection rather than delivering incorrect messages.
8.5.1	EMS-7693	Fixed a defect where a pair of fault-tolerant EMS servers set to start with a JSON configuration, obtained by running the tibemsdft-*.conf sample files through the tibemsconf2json utility, would not fail over properly. This defect affected Release 8.5.0, on the AIX, HP-UX, Solaris and z/Linux platforms only.
Issues Closed in Release 8.5.0		
8.5.0	EMS-7650	Fixed a defect where a Java client, on which the verification of the name in the CN field of the server certificate has been disabled, may have this verification re-enabled on a reconnect. This could result in a failure to reconnect. This defect affected Releases 8.4.0 and 8.4.1.
8.5.0	EMS-7636	Fixed a defect that affected Java clients performing SSL connections on IBM platforms. If a client identity was not specified, and a system level trust file contained a key, that key would be used when initializing SSL, and erroneously, the corresponding client identity would be presented to the server. This defect affected Releases 8.4.0 and 8.4.1.

Closed in Release	Key	Summary
8.5.0	EMS-7626	Fixed a defect where a compact operation on a file store could temporarily introduce an inconsistency in the server state persisted on disk for a limited window of time. If the server was shut down or failed over before that window elapsed, the state recovered at the next startup could be stale, resulting in unpredictable behavior such as the redelivery of already delivered messages or transaction errors. This defect affected Releases 8.3.0, 8.4.0 and 8.4.1.
8.5.0	EMS-7582	Fixed a defect that caused the EMS server to leak a socket created against an SSL listen when that socket subsequently timed out. This could happen, for example, with non-EMS client incoming connections on an SSL listen or during large bursts of incoming SSL connections.
8.5.0	EMS-7556	Fixed a defect where a Java client would throw an exception during an SSL connection attempt if there was no system level trust file, even though the client provided its own trusted certificates. This defect affected Releases 8.4.0 and 8.4.1.
8.5.0	EMS-7553	Fixed a defect in which the `network_thread_count` was effectively decreased by one after the first fault-tolerance failover of a pair of EMS servers.
8.5.0	EMS-7532	Fixed a defect where shared durable consumers are turned into shared non-durable consumers upon server restart or fault-tolerance failover.
8.5.0	EMS-7531	Fixed a defect where previously deleted shared subscriptions may return upon server restart or fault-tolerance (FT) failover. Because these stale shared subscriptions may have out-of-date properties, this could result in reconnecting FT consumers receiving messages from the wrong destination or evaluated with a wrong selector.
8.5.0	EMS-7509	Fixed a defect where a route would be shown as persisted in a local configuration file after a failed route update through the administration tool even though it had not been persisted.
8.5.0	EMS-7506	Fixed a defect where the server could fail to properly close a consumer if the corresponding client API call was invoked during or just prior to a Fault-Tolerance failover. This could result in a number of messages being retained by the server until the underlying session was closed.
8.5.0	EMS-7504	EMS could fail to properly close a session in the server if the corresponding C or .NET client API call was invoked during a Fault-Tolerance failover and using an SSL connection. This has been fixed.
8.5.0	EMS-7499	Fixed a defect in Central Administration where selecting the improperly labeled **Order by Subscribers** drop-down option in the Topics Monitoring page did not have any effect. It is now called **Order by Subscriptions**.

Closed in Release	Key	Summary
8.5.0	EMS-7492	When you grant an administration permission that includes other permissions and later revoke one of the permissions it includes, this internally translates into one positive and one negative permission, such as: `view-all,-view-message`. Fixed a defect where negative permissions set through the administration tool would not be persisted when the EMS server was using a JSON configuration file.
8.5.0	EMS-7470	Fixed a defect in Central Administration that caused the **Verify Host** and **Verify Hostname** check boxes presented on various pages to always revert to 'checked' upon reload of the page and to set the corresponding configuration properties to an invalid value.
8.5.0	EMS-7459	Fixed a defect where the value for the `trace_client_host` server property would be switched to `both` upon any update made to the main `tibemsd.conf` file through the administration tool or Admin API if that value had initially been set to `both_with_port`.
8.5.0	EMS-7434	Fixed a defect in Central Administration where unchecking the **Message Swapping** check box would forcefully reset and hide the **Maximum Message Memory** property.
8.5.0	EMS-7371	Fixed a defect in which the value set through the `ldap_tls_cacert_dir` property would not be picked up when using a JSON configuration file. This defect affected Releases 8.3.0, 8.4.0, and 8.4.1.
8.5.0	EMS-7336	When Fault Tolerance (FT) is configured for using SSL in between servers, each server with its certificate populated with its own CN field, the **Secondary Expected Hostname** property available on the Fault Tolerance page of Central Administration can be used to configure the FT expected hostname of the secondary server. Fixed a defect where that **Secondary Expected Hostname** property was mistakenly also available on the Routes and Factories pages, where it was irrelevant and ineffective.
8.5.0	EMS-7321	Fixed the C sample programs to return 0 upon successful completion.
8.5.0	EMS-7288	Fixed a defect in which the sample certificates installed with EMS wouldn't work when used with .NET client programs. This defect affected Releases 8.4.0 and 8.4.1.
8.5.0	EMS-6772	Fixed a defect which caused the `FTL` trace option to be omitted when displaying the list of currently enabled server log or console traces.
Issues Closed in Release 8.4.1		
8.4.1	EMS-7471	Fixed a defect where the server would on rare occasions not fully restore a shared durable subscription from the `$sys.meta` store upon server restart or fault-tolerance failover. In such a case, although that subscription would eventually work properly, it was unable to deliver or retain messages for some time. Unshared durable subscriptions were unaffected.

Closed in Release	Key	Summary
8.4.1	EMS-7461	Fixed a defect where, in rare situations, a durable consumer set on a global topic could stop receiving messages over a route if the remote server was restarted or went through a fault-tolerance failover.
8.4.1	EMS-7456	Fixed a defect in which calling `show transaction` in the administration tool or its Admin API equivalent could cause a memory leak and an incorrectly larger Pending Messages count and Pending Message Size being reported in `show server` until the server was restarted.
8.4.1	EMS-7447	Fixed a defect in which closing a queue consumer or a durable consumer that was involved in an XA transaction between the XA end and the XA prepare resulted in the messages already received by that consumer in that transaction being put back on the destination. The messages would then be available to other consumers. A subsequent XA commit or rollback could result in a transaction exception. In such a situation, the server will now prevent those messages from being available until an XA rollback. XA commit or rollback won't result in that same transaction exception anymore.
8.4.1 8.4.0-CE	EMS-7425	Fixed a defect in which the separate counts for client and admin connections reported by the EMS server since Release 8.4.0 could be incorrect when using LDAP authentication if the LDAP server was slow.
8.4.1 8.4.0-CE	EMS-7424	Reverted the EMS-7330 fix due to a risk of store file corruption it had introduced.
8.4.1 8.4.0-CE	EMS-7397	Fixed a defect in which the detection of a duplicate message sent within a transaction could cause the server to crash if `track_message_ids` was enabled. In such a situation, the server will now trace the message and roll back the corresponding transaction.
8.4.1 8.4.0-CE	EMS-7392	Fixed a defect in the client library that affected Java clients with multiple SSL connections to the server. If a Java client concurrently attempted to establish multiple SSL connections, some of the connection attempts could fail. This symptom could also occur while reconnecting after a server failover. This symptom could occur in Release 8.4.0 Java clients.
8.4.1 8.4.0-CE	EMS-7379	Fixed a defect in which very large server trace statements would be missing their line break.
8.4.1 8.4.0-CE	EMS-7373	Fixed a defect in which the `$sys.meta` file-based store could grow indefinitely. This symptom could occur after clients had used fault-tolerant connection URLs. This defect affected hotfix 8.4.0_HF-003 only and is not relevant since EMS-7424.
8.4.1 8.4.0-CE	EMS-7366	Improved trace messages related to slow operations.
8.4.1 8.4.0-CE	EMS-7365	Improved performance by servicing client connection requests in a dedicated thread.

Closed in Release	Key	Summary
8.4.1 8.4.0-CE	EMS-7358	Fixed a defect in which deleting a static destination could erroneously clear permissions from other destinations. This symptom could occur only when the server was configured using a JSON file.
8.4.1 8.4.0-CE	EMS-7356	Fixed a defect which could stop communication between the EMS server and an LDAP server.
8.4.1 8.4.0-CE	EMS-7330	Fixed a defect in which clients could miss messages or receive duplicate messages. This symptom could affect messages within transactions that were persisted to a file-based store on an NFSv4 file system. Network disconnects could trigger this symptom. This fix has been reverted by EMS-7424.
8.4.1 8.4.0-CE	EMS-7323	Fixed a defect in which the reconnection of a client after a fault-tolerance failover could cause another connection that was given the same ID in the meantime to be terminated. This could occur under rare circumstances and manifest in the server log as "reconnect detected: replacing active connection". This defect affected Release 8.4.0.
8.4.1 8.4.0-CE	EMS-7296	Fixed a defect in which slow operations could interfere with the heartbeat protocol between the EMS server and clients, resulting in client disconnect. This defect affected Releases 8.3.0 and 8.4.0.
Issues Closed in Release 8.4.0		
8.4.0	EMS-7141	Previously, an attempt to view JNDI names was refused by the server if the requesting user did not have the `view` permission on the underlying destinations. Even so, the server would return a normal response with no JNDI names present. This has been fixed so that the admin client now receives a response indicating that the attempt was refused due to a lack of permissions.
8.4.0	EMS-7131	On AIX, Linux, z/Linux and Solaris, the server would crash upon handling a JSON configuration that held UTF-16 characters. This has been fixed so that the server will continue running, rejecting the JSON configuration when appropriate. UTF-16 is not supported by EMS in JSON configuration files.
8.4.0	EMS-7101	The number of parameters accepted by the `tibemsca.bat` and `tibemsconf2json.bat` scripts on Windows platforms was not unlimited. This has been fixed.
8.4.0	EMS-7046	Setting the `server_rate_interval` server property to 1000 through the administration tool or API on a JSON-based server would result in that value not being persisted in the configuration file. Upon restart, the server would then use the default value of 1. This has been fixed.
8.4.0	EMS-7031	Starting with Java 8 update 91 or higher, Boolean variables passed from the EMS server to its embedded JVM would always be read as `false`. It would affect features that rely on that JVM such as JAAS and `dbstores`. This has been fixed.

Closed in Release	Key	Summary
8.4.0	EMS-7023 EMS-7022	A possible deadlock could prevent a server from starting up or a standby server from activating upon recovering messages for a topic with `maxbytes` or `maxmsgs` set. This has been fixed.
8.4.0	EMS-7017	Fixed an error that caused a memory leak when the server was initializing `mstores`.
8.4.0	EMS-7016	Used in the 64-bit C client, the `tibemsSession_GetAcknowledgeMode` function could return an incorrect value or cause a crash on Solaris and macOS. This has been fixed.
8.4.0	EMS-6991	Messages sent with a non-zero delivery delay are temporarily stored in a system queue prefixed with `$sys.delayed`. When the delay has expired, the server moves them to their target destination. If the server was unable to do that because `max_msg_memory` was exceeded, those messages would remain on the `$sys.delayed` queue permanently. This has been fixed.
8.4.0	EMS-6981	If changes were made directly to the EMS server configuration using the Administration Tool or the Admin API and the corresponding configuration in Central Administration was not refreshed, the deployment was rejected. In that context, the `Full EMS server deployment log available here` link available on the Deployment Server Log panel showed a Java stack trace instead of a proper deployment log. This has been fixed in the EMS server.
8.4.0	EMS-6758	EMS servers using JSON configuration files did not enforce parameters with values set to numbers with 7 or more digits through the administration tool or admin API. This has been fixed.
8.4.0	EMS-6756	Fixed an issue that could cause an EMS server to crash if Central Administration was used to deploy an empty value for the `ldap_tls_cert_file` property.
8.4.0	EMS-6752	Previously, the .NET client did not support TLS 1.2, although the C and Java clients did. This has been fixed.
8.4.0	EMS-6750	Fixed an issue that prevented the administration tool from setting an empty server password on a server that is using a JSON configuration file.
8.4.0	EMS-6746	Upon receiving an incoming SSL connection, an EMS server configured to use a JAAS module passes user information to that module, including SSL data. In the case of routes using SSL, the server would fail to pass the SSL data to the JAAS module, which may require it depending on its implementation. This has been fixed.
8.4.0	EMS-6739	Fixed an issue that would cause the EMS server to crash when setting the `client_trace` server property with a `user=` filter.

Closed in Release	Key	Summary
8.4.0	EMS-6736	A route protocol error has been fixed that could prevent the delivery of messages from routes across topic-to-topic bridges. For the error to occur, three or more routed servers would need to be sharing a global topic bridged to a local topic. A subset of consumers on the local topics could then fail to receive messages sent over the source global topic.
8.4.0	EMS-6735	In earlier versions of EMS, the connect/reconnect system properties such as `tibco.tibjms.connect.attempts` directly affected the `TibjmsConnectionFactory` class used internally by a `TibjmsUFOConnectionFactory`. This resulted in unexpected connection/reconnection behavior when using unshared state failover. Also, values set using those system properties were not reflected in the output of the client trace. This has been fixed. The `TibjmsUFOConnectionFactory` class now directly controls the connect/reconnect behavior of the `TibjmsConnectionFactory` that it uses internally. Furthermore, it now uses the connect/reconnect system properties to control its connections. If need be, you can revert to the previous behavior.
8.4.0	EMS-6733	The `Slow clock tick` log message has been adjusted to be more specific as to how the EMS server was affected. The message now states that either connection timeouts were delayed or messaging *and* connection timeouts were delayed due to an overloaded machine.
8.4.0	EMS-6732	On Linux platforms, EMS software release 8.3 introduced a change in how the EMS server opens store files that prevents the file's last access time from being updated when the file is read. This may improve performance but has the adverse effect of requiring the user ID of the server process to match that of the owner of the store file. Violating this requirement could prevent the server from starting. This has been fixed: upon failing to open a store file in such circumstances, the server now tries to open it again the standard way.
8.4.0	EMS-6730	If you attempt to rename an already deployed queue, topic, or durable, Central Administration, you will now be warned that this would result in that item being deleted and recreated upon deployment.
8.4.0	EMS-6722	The cipher suites used by Central Administration for its own SSL connections could not be specified. This can now be done through the new `--ssl-ciphers` option of the Central Administration server. It is applicable to both the connections to EMS servers when using SSL and the incoming Web browser connections when using HTTPS.
8.4.0	EMS-6713	Fixed an issue on UNIX platforms that would prevent an EMS server from starting up if it had failed to lock a file on the first attempt.
8.4.0	EMS-6710	Fixed an issue that could cause the server to report corrupted records on startup for file-based stores with `file_truncate` enabled and `file_minimum` set with a value. Note that no valid data was corrupted (only previously freed records), so no message loss would occur.

Closed in Release	Key	Summary
8.4.0	EMS-6700 EMS-6741	EMS could fail to properly close a session in the server if the corresponding Java client API call was invoked during a fault tolerant failover and using an SSL connection. This has been fixed.
8.4.0	EMS-6698	Previously, in Central Administration dynamic topics were not displayed in the list of monitored topics using the same notation as for dynamic queues. This has been fixed. An asterisk (*) now appears in front of dynamic topics, as it does for dynamic queues.
8.4.0	EMS-6694	Fixed an issue where refreshing an EMS server configuration in Central Administration would leak an admin connection in the Central Administration server. This has been fixed.
8.4.0	EMS-6690	Upon establishing an SSL connection, if both a client and the server use each their own certificate with an extended key usage extension that includes the client authentication flag, the connection should fail. Previously, this would be the case when trying to connect with a C client, but a Java client would connect successfully. This has been fixed: under such circumstances, a Java client will no longer be allowed to connect.
8.4.0	EMS-6689	Previously, when authentication of the server by a client required a chain of trusted certificates (e.g. a top and a mid level certificate), the authentication succeeded when providing just the top level trusted certificate for the C client whereas both the top level and the mid level certificates were required for the Java client. This discrepancy has been fixed: the Java client can now authenticate the server by providing only the top level trusted certificate.
8.4.0	EMS-6686	Previously, when EMS clients used certificates with an extended key usage extension, they were required to include the client authentication flag. This has been fixed. Now, client certificates will be considered valid with either the client authentication flag or the server authentication flag, or both.
8.4.0	EMS-6684	Fixed an issue that would cause the EMS server to crash when deleting a route with an invalid protocol in the URL.
8.4.0	EMS-6680	Fixed an issue that would cause the EMS server to crash due to an invalid session when in a dual-active situation.
8.4.0	EMS-6677	The maintenance and the build versions of the server were previously not shared between servers in a fault-tolerant pair. This has been fixed so the full EMS version is correctly displayed when viewing a fault-tolerant connection through the administration tool, API, or Central Administration.
8.4.0	EMS-6665	Beginning with release 8.3, the addition or the deletion of stores of type file made through Central Administration no longer required restarting the standby server of a fault-tolerant pair of EMS servers. Restarting the active server was enough. However, modifications made to the properties of an existing store of type file were still not effective in the standby server until it had been restarted. This has been fixed: these modifications will now be picked up by the standby server. This has been fixed.

Closed in Release	Key	Summary
8.4.0	EMS-6663	Fixed an issue where any stores configured to bind to a large and unavailable processor ID could cause an EMS server on Linux to crash at startup.
8.4.0	EMS-6662	The documented `com.tibco.tibjms.ssl.cipher_suites` Java client property was being ignored, whereas the undocumented `com.tibco.tibjms.ssl.ciphers` property did work. This has been fixed. They now are documented synonyms. If both are defined, the latter overrides the former.
8.4.0	EMS-6659	Multiple bridge definitions with the same source destination are grouped into single entries when viewed through the administration tool, but were not grouped when viewed through Central Administration. This has been fixed. Central Administration has been adjusted to provide the same feature, along with persisting the grouped entries to a server's JSON configuration file upon the next deployment.
8.4.0	EMS-6641	In Central Administration, attempts to purge or destroy durables that have a client ID would fail. This has been fixed.
8.4.0	EMS-6634	Previously, Central Administration failed to validate for mutually exclusive destination properties. It now flags a validation error when a global or routed queue is marked exclusive or prefetch is set to `none`. It also flags an error when a routed queue has a `Redelivery Delay` set.
8.4.0	EMS-6616 EMS-6725	The EMS server could leak sockets when an incoming connection failed (for example, because of a network failure). This has been fixed. Incoming connection handshakes are now subject to a timeout that is the larger of `server_timeout_server_connection` and `server_timeout_client_connection`, if either is specified. Otherwise, the timeout is `handshake_timeout`, if specified. If none of these are specified, the timeout on the incoming connection handshake defaults to 3 seconds.
8.4.0	EMS-6500	The default TCP send and receive buffer sizes are now determined on all platforms by the underlying operating system. This behavior was previously available only on Linux. If needed, you can still use the `socket_send_buffer_size` and `socket_receive_buffer_size` parameters to manually set the TCP send and receive buffer sizes.
8.4.0	EMS-6494	In Central Administration, attempting to roll back a transaction gave the error `Unable to find transaction`. This has been fixed. Transactions can now be rolled back through Monitoring > Transactions > *select a transaction* > Manage transaction > Rollback. Also, after a transaction rollback has succeeded, the transaction page is reloaded to list only the pending transactions.
8.4.0	EMS-6487	Fixed an issue that would cause the `show stat producers queue=`*name* administration tool command to include non-matching producers created before the `detailed_statistics` server property was set to `PRODUCERS`.

Closed in Release	Key	Summary
8.4.0	EMS-6436	Modified the filter used on Central Administration pages that display lists of destinations to use destination wildcards. These pages include Topics, Queues, Monitoring Topics, and Monitoring Queues.
8.4.0	EMS-6382	In Central Administration, the filtering of consumers on the Monitoring > Consumers page was incomplete. This has been fixed. You can now filter consumers by Consumer ID, User, Destination, or Type.
8.4.0	EMS-6121	The host name of the machine where the EMS server runs was missing from the server information provided by the administration tool and API and Central Administration. This has been fixed.
8.4.0	EMS-6120	When connected using a fault-tolerant URL, the administration tool would fail to update the URL in its prompt after a failover. This has been fixed.
8.4.0	EMS-5240	Mutual authentication of a client and the server with FIPS compliance enabled would fail when using mixed DSA and RSA certificates, that is to say with the client using a DSA certificate and the server an RSA certificate or the other way around. This has been fixed.
8.4.0	EMS-4105	When using the IBM JVM on AIX or z/Linux, the Java client failed to support certificates in the PEM format if their file name ended in `.p7`. This has been fixed.
8.4.0	EMS-3049	When running EMS with FIPS compliance enabled, DSA certificates could not be used. This has been fixed.
Issues Closed in Release 8.3.0		
8.3.0	EMS-6644	In previous releases, the SmartSockets `T_MSG_FT_CHAR` type was converted to an EMS `Byte` type while the EMS `Character` type was converted to the SmartSockets `T_MSG_FT_INT2` type. With EMS 8.3, the latter doesn't change but a SmartSockets `T_MSG_FT_CHAR` type is now converted to an EMS `Character` type.
8.3.0	EMS-6625 EMS-6617	Some applications may populate the `JMSReplyTo` header field of a message using a destination that has a `NULL` name. This may cause the EMS server to crash under particular circumstances. The C client would also crash upon receiving such a message. This has been fixed.
8.3.0	EMS-6623	Fixed a memory leak associated with the C function `tibemsMsg_MakeWriteable`.
8.3.0	EMS-6622	The servers in a fault tolerant pair of EMS servers set with SSL listen ports can be configured to use SSL certificates that have different CN names matching their respective hostnames. In this particular case, an issue prevented Java clients from reconnecting to the new active server in a failover situation. This has been fixed.

Closed in Release	Key	Summary
8.3.0	EMS-6613	After it had rejected an invalid deployment, the EMS server would also reject a subsequent deployment even if that one was valid, incorrectly asking the Central Administration user to refresh the configuration and thereby lose their changes. This has been fixed.
8.3.0	EMS-6511	To enhance security, EMS Central Administration can no longer be presented in the frames of other web pages.
8.3.0	EMS-6502	Fixed an issue where a deployment through Central Administration would crash an EMS server when the deployment involved enabling the `trace` property on a routed queue.
8.3.0	EMS-6499	Fixed an issue in Java clients that would cause the one invalid cipher suite in a list of several to invalidate the whole list. The Java client, like the C client, now skips the invalid cipher suite rather than throwing an exception.
8.3.0	EMS-6459	Only on AIX, the `libldap` shared library that is included with EMS had a dependency issue. That issue could result in a runtime error for EMS C client programs, such as: `0509-150   Dependent module (...)/liblber.so could not be loaded.` This has been fixed.
8.3.0	EMS-6404	Previously, the pre-built JAAS modules for LDAP authentication would allow anonymous binds (correct username with no password provided) if the backing LDAP server was an ActiveDirectory LDAP server that supported simple binds. This has been fixed such that the JAAS modules will explicitly reject anonymous bind attempts, in line with the behavior of the original implementation of LDAP authentication within the EMS server.
8.3.0	EMS-6367	Using the / UNIX path separator in the `logfile` server property when `logfile_max_size` was also set would cause log file rotation to fail on Windows. This has been fixed. On UNIX systems, a warning is now logged when the `logfile` property contains the \ Windows path separator.
8.3.0	EMS-6356	Fixed a problem that would cause the EMS server to reject an incoming connection attempt if the expected number of bytes was not received in one packet during initial handshake. This could be caused by proxies or load-balancers.
8.3.0	EMS-6352	The modification of the `console_trace` or `log_trace` server properties through the Administration Tool or Admin API was not working properly when using a JSON configuration file. This has been fixed.
8.3.0	EMS-6339	Fixed an issue that could cause an application using the C client library to crash when failing to read a compressed message due to out-of-memory conditions. The library will now correctly return the TIBEMS_NO_MEMORY status.

Closed in Release	Key	Summary
8.3.0	EMS-6300	Fixed an error where the default values for Verify Host and Verify Hostname in Central Administration were incorrect on the Fault Tolerance, Factories and Routes pages.
8.3.0	EMS-6299	The server has a limit of 32,767 string literals in an IN clause of a selector. The client libraries now prevent the creation of consumers with selectors that do not satisfy this limit.
8.3.0	EMS-6297	The EMS server would accept a Central Administration deployment even if it included a duplicate store file name. This has been fixed: the EMS server now rejects such a deployment.
8.3.0	EMS-6295	The EMS Server now correctly fails the creation of a consumer with selector if any error occurs during the processing of this selector.
8.3.0	EMS-6282	Externally defined users who are members of the `admin` group are now able to successfully subscribe to system monitoring topics when authorization is disabled, while before the user's subscription would have been refused.
8.3.0	EMS-6281	Previously, Central Administration would display the Use CRC to Validate File Integrity File Store property and the Multicast Enabled factory property incorrectly when they were set to the default value. This has been fixed.
8.3.0	EMS-6280	Previously, the User Certificate Spec Name field of the SSL page in Central Administration would be mistakenly hidden under some circumstances. This has been fixed.
8.3.0	EMS-6268	Fixed an issue that could cause store statistics to show negative message size when queue messages were moved to the undelivered queue after reaching the maximum redelivery count.
8.3.0	EMS-6267	When they held some characters such as `$`, SSL passwords were sometimes encoded incorrectly by the Central Administration server, resulting in the EMS server being unable to decode and make use of those passwords. This has been fixed.
8.3.0	EMS-6262	The client-side tracing of connections did not include the relevant connect and reconnect property values when using unshared state connections. This has been fixed by the addition of a separate trace line labeled `UFOConnectionFactory`.
8.3.0	EMS-6245	Fixed an issue that would cause the Message Memory Usage to be much higher than expected when sending compressed messages to a destination with the `sender_name` or `sender_name_enforced` properties and message swapping disabled.
8.3.0	EMS-6238	Fixed an issue that could prevent a server from connecting to its fault tolerant or routed peer.
8.3.0	EMS-6220	Fixed an issue that could cause the EMS Server to crash when creating a route or a bridge with a selector syntax error.

Closed in Release	Key	Summary
8.3.0	EMS-6218	Fixed an error that caused a shortcut to the EMS Administration Tool to be created on Windows, even when the actual executable has not been installed.
8.3.0	EMS-6216	On Windows and Linux, the server, C clients, and Administration Tool, when given an URL with `localhost` (or no hostname), now rely on the name resolution of `localhost` instead of trying to connect to IPv6 and IPv4 loopback addresses (::1 and 127.0.0.1).
8.3.0	EMS-6215	If the respective values for the `logfile` properties of two EMS servers in a fault tolerant pairs were different, the `tibemsconf2json` tool would not take that into account. This has been fixed: the resulting JSON configuration file will now include the corresponding `secondary_logfile` property.
8.3.0	EMS-6211	Fixed an issue that would cause admin permissions granted to an external user that was connected to the server to be lost when that external user disconnected. Note that permissions would still be persisted in the ACL configuration.
8.3.0	EMS-6204	Central Administration offers the ability to compact a store through the *server name* > Monitor > Stores > store name > Manage Stores > Compact store… option. Before initiating the compaction, Central Administration shows a warning stating that the operation will time out after 120 seconds. However, the timeout effectively used was 60,000 seconds (16 hours and 40 minutes), during which time all other EMS server operations were suspended. This has been fixed.
8.3.0	EMS-6201	Fixed an issue that would prevent the creation of a route when the `incoming_topic` property was specified with a topic name that was not present in the configuration.
8.3.0	EMS-6199	If a consumer on a routed queue received a message and exited before acknowledging it, the `JMSXDeliveryCount` was not incremented. This has been fixed.
8.3.0	EMS-6195	Fixed an error in Central Administration that prevented Client Trace from being shown as checked, even when its value in the JSON configuration file was "enabled".
8.3.0	EMS-6194	Fixed an issue that could cause failure to connect to a server if the hostname specified in the URL resolved to both an IPv4 and IPv6 addresses, but the IPv6 address was disabled.
8.3.0	EMS-6147	In Central Administration, the FTL Realm Server URL Secondary was not visible on the Transports page after it had been deployed. This has been fixed.
8.3.0	EMS-6104	Fixed an error that caused a memory leak when the server was loading its configuration if it was in the JSON format.

Closed in Release	Key	Summary
8.3.0	EMS-6095	Fixed an error that caused a memory leak if the server encountered an issue upon creating an FTL transport.
8.3.0	EMS-6076	The EMS server allowed configurations where a wildcard destination would import a transport of type TIBCO FTL, which is invalid. This has been fixed.
8.3.0	EMS-6059	The EMS server allowed configurations where multiple destinations would import the same TIBCO FTL transport, which is invalid. This has been fixed.
8.3.0	EMS-6039	When given a deployment name with invalid characters, Central Administration would show a `Bad resource name` error. This has been improved with clearer instructions.
8.3.0	EMS-6014	Fixed an error that could cause a memory leak in a standby EMS server upon activation if a TIBCO FTL transport was configured.
8.3.0	EMS-5858	Compressed messages on a destination with `sender_name` or `sender_name_enforced` properties that have been sent using XA transaction and left in prepared state may become available after a server restart without an XA commit. This has been fixed.
8.3.0	EMS-5817	Fixed an issue that would prevent adding/modifying/removing properties from a global queue if its name was of the form *queue-name*@*server-name* and *server-name* was the home server. This was an issue only when using a JSON configuration file.
8.3.0	EMS-5752	Even though it should be allowed, closing an unshared state session object after the corresponding unshared state connection had been closed led to an exception. This has been fixed.
8.3.0	EMS-5383	On Windows, the EMS installation did not hold the correct path to the sample configuration in the **Start EMS Server** shortcut. This has been fixed.
8.3.0	EMS-4498	Fixed an issue that would prevent the EMS server from detecting a duplicate store file name if the default store (`$sys.meta`, `$sys.nonfailsafe` or `$sys.failsafe`) was not configured, but its file name (`meta.db`, `async-msgs.db` or `sync-msgs.db`) was used by a user-defined store.
8.3.0	EMS-3581	Fixed an error that permitted the use of a temporary destination as part of a bridge.
8.3.0	EMS-2837 1-AC2L2T	On AIX, HP-UX and Solaris, LDAP authentication could fail in some situations when the undocumented `ldap_operation_timeout` server property was set. This has been fixed on all platforms.
8.3.0	EMS-2332	Fixed an issue that would lead an EMS server in an FT pair configured with SSL for FT to incorrectly log `SSL handshake failed: ret=-1, reason=<unknown>` at startup if its peer was not yet present.

Closed in Release	Key	Summary
Issues Closed in Release 8.2.2		
8.2.2	EMS-6236	Fixed an issue that could cause the following error message: `Failed writing message to 'store-file-name': I/O error or out of disk space.` This could happen when moving compressed persistent queue messages on synchronous stores to the undelivered queue.
8.2.2	EMS-6226	The EMS server crashed when it accepted too many client connections concurrently. This has been fixed.
8.2.2	EMS-6165	Under particular circumstances involving long network round-trips, LDAP authentication would fail intermittently. This has been fixed.
8.2.2	EMS-2521	Fixed an issue that could cause the EMS server to slow down when messages were expiring and a large number of messages were held by the server.
Issues Closed in Release 8.2.1		
8.2.1	EMS-6192	Fixed an issue that could cause the EMS server to crash when the `logfile` or `ssl_crl_path` parameters were specified and the directory contained long file names.
8.2.1	EMS-6182	The SSL facilities of the EMS 8.2.0 C client are not forward-compatible with servers in future EMS releases. This has been fixed with the EMS 8.2.1 C client.
8.2.1	EMS-6181	Fixed an issue that would cause the server to reject incoming messages (that have a message and/or correlation ID) when enabling "Track Message IDs" and/or "Track Correlation IDs" from Central Administration.
8.2.1	EMS-6180	Previously, the EMS server would reject a deployment from EMS Central Administration if it detected that the configuration was changed using the tibemsadmin tool or admin API until the administrator refreshed the configuration. However, if a failover occurred, the newly active EMS server would fail to detect this situation and accept a deployment that should have required a refresh. This has been fixed.
8.2.1	EMS-6179	Changing server properties that do not require a restart (such as authorization) on the active server of a fault tolerant pair was not reflected in the runtime state of the standby server after activation. This has been fixed in the case when EMS servers use JSON configuration files.
8.2.1	EMS-6178	Fixed a problem that could cause the EMS server state to be reported incorrectly to admin clients connecting to an EMS Appliance. When the server was in the `wait-for-peer` state, the `State.get()` admin API call returned `11` instead of returning the `State.SERVER_STATE_WAIT_FOR_PEER` convenience constant.

Closed in Release	Key	Summary
8.2.1	EMS-6171	In EMS 8.2.0, it was possible that a JMS consumer equipped with a queue selector that used the JMSCorrelationID, JMSMessageID or JMSType header fields and that did not use any JMS properties would not receive the corresponding messages if these could be swapped to disk. This has been fixed.
8.2.1	EMS-6142	Fixed an issue that could cause a standby EMS server that activated on failure of the active server to abruptly exit when processing message acknowledgments.
Issues Closed in Release 8.2.0		
8.2.0	EMS-6093	Fixed an issue that could prevent an EMS server from honoring the `max_connections` limit. This occurred for example when fault tolerant clients reconnected due to network issues, or during the purge of connections after a server restart and the resulting fault tolerant reconnect timeout.
8.2.0	EMS-6068	Fixed an issue that could cause an EMS C client application to crash when two threads called `tibemsConnection_Close()`, on the same connection, at the same time.
8.2.0	EMS-6063	Fixed an issue that could cause topic subscribers with selectors and/or bridge targets with selector to stop receiving messages.
8.2.0	EMS-6036	When a route was promoted from passive to active using the administration tool or the admin API with a JSON-configured EMS server, that operation would succeed but the change was not persisted into the JSON file. (The change would be persisted if it was made using EMS central administration instead.) This has been fixed.
8.2.0	EMS-6027	Fixed an issue that would cause messages on a Shared Non-Durable Subscription to not be redelivered immediately.
8.2.0	EMS-6024	Fixed an error that could prevent the EMS server from exiting when a store file encountered a non-retryable write error.
8.2.0	EMS-6021	Fixed an issue that sometimes prevented queue messages from being immediately redelivered following a roll back.
8.2.0	EMS-6001	Previously, the `JMSDeliveryTime` for messages imported from Rendezvous or SmartSockets was not set and defaulted to zero instead of its correct value. This has been fixed.
8.2.0	EMS-5873	Fixed an error that caused the `tibemsconf2json` tool to truncate the target destination of a bridge if that destination had the word `selector` in its name.
8.2.0	EMS-5862	Previously, the text of the error generated when a client failed to authenticate with the EMS server could be misleading. It has been replaced with a more generic text: `authentication failed`.

Closed in Release	Key	Summary
8.2.0	EMS-5851	When creating a new store in Central Administration, the store type for `mstore` was mistakenly displayed as `Multiple store` in the `type` dropdown box. This has been corrected to show the store type as `mstore`.

8.2.0	EMS-5849	Fixed an issue that could potentially delay clients' clock synchronization when the server parameter `clock_sync_interval` was specified and clients' connections were closed or lost. The following warnings could also be produced:
		
		`WARNING: Clock sync timer error: Not Found`
		`WARNING: Clock sync timer error: Invalid Argument`

8.2.0	EMS-5846	Fixed an issue that could prevent the destination's expiration override property from being honored in certain situations.
8.2.0	EMS-5841	Previously, a message selector configured on a bridge and using the `JMS_TIBCO_SENDER` message property would fail to select the corresponding messages. This has been fixed.
8.2.0	EMS-5840	Conditions leading to the truncation of file stores have been tweaked to be more predictable.
8.2.0	EMS-5837	The effect of setting the connect attempt and reconnect attempt properties at the client level on applications that use unshared state connection factories was not documented. You can now refer to the Set Connect Attempt and Reconnect Attempt Behavior in the *TIBCO Enterprise Message Service User's Guide* for more information.
8.2.0	EMS-5834	Fixed an error that caused unexpected results when the same EMS server URL was repeated multiple times within the unshared state configuration settings. For example, if an unshared state client used a URL of the form Server_A+Server_A+Server_B and Server_A was down, the UFO client never connected to Server_B. In this scenario, the unshared state client will now connect to Server_B.
8.2.0	EMS-5825 EMS-5842	Fixed an issue that could cause redelivered messages from a session with `DUPS_OK_ACKNOWLEDGE` or `EXPLICIT_DUPS_OK_ACKNOWLEDGE` to have the `JMSRedelivered` flag set to `false` instead of `true` after the closing of a consumer.
8.2.0	EMS-5823	Fixed an error that could cause an application to crash with a First-chance exception when using the Windows `LoadLibrary` function to load EMS DLLs, if no subsequent API call was made.
8.2.0	EMS-5820 EMS-5821	Fixed a problem that could cause deadlock in the client libraries when closing a session (with a previously closed durable consumer with unacknowledged messages) in one thread, and acknowledging messages from a durable consumer created with another session in another thread. Those sessions came from the same connection.

Closed in Release	Key	Summary
8.2.0	EMS-5812	In EMS 8.0 and 8.1, it was not possible to add a new EMS server to the Central Administration server through a SSL connection URL. This has been fixed.
8.2.0	EMS-5811	Fixed an issue that prevented a routed queue consumer from receiving messages if the queue's name in the home server referenced its own server. For example, if the configuration of server EMS-SERVER contained the global queue myQueue@EMS-SERVER. When the routed queue consumer was started on a proxy server, the following warning message would appear on the home server: `WARNING: Routed Queue 'myQueue' is not a home Queue`
8.2.0	EMS-5806	Fixed an issue that could cause the server to crash when a combination of events regarding a durable consumer occurred. Multiple situations could cause this, but all scenarios have the following events in common: • Closing a durable with unacknowledged messages • Re-opening (and later closing) the durable using a different session • Unsubscribing the durable subscription Crashes could occur in different places, including but not limited to consumers statistic gathering, dynamic destination cleanup, session recover, and so on.
8.2.0	EMS-5804 EMS-5857	Fixed a possible deadlock in the client libraries. Situations where the client library could deadlock included when a connection was started, stopped or closed, and when a session was created. The risk of a deadlock was increased when the clock synchronization feature was used (`clock_sync_interval` defined in the server), or when messages had a `JMSExpiration` set.

8.2.0	EMS-5776	The following parameters have been added to the output of the show config command in the administration tool:
		`processor_ids`
		`network_thread_count`
		`selector_logical_operator_limit`
		`max_msg_print_size`
		`max_msg_field_print_size`

8.2.0	EMS-5775	Fixed an error that could cause the EMS server to crash when it had more than 32,000 connections.
8.2.0	EMS-5756	An ALL server tracing option was mentioned in comments of the sample server configuration files when such a tracing option does not exist. This has been fixed.
8.2.0	EMS-5732	Fixed an issue that prevented a SSL trusted certificate from being correctly added into a `ssl_trusted_list` or a `ssl_issuer_list` in the EMS JSON configuration file, when added through the administration tool.

Closed in Release	Key	Summary
8.2.0	EMS-5514	Previously, an EMS server with an active SSL route showed the corresponding connection as non-SSL. There was a similar issue with the connections between two EMS servers in a fault tolerant pair, if using SSL. This has been fixed.
8.2.0	EMS-5374	Fixed an error that could cause a C client to crash if a session was closed before closing a queue browser that was created using that same session.
8.2.0	EMS-4752	Previously the Unix scripts `tibemsd.sh` and `tibemsd64.sh` did not have executable permissions and could not be invoked from other directories. In EMS 8.2.0, `tibemsd.sh` and `tibemsd64.sh` are now installed with executable permissions and can be invoked from any directory.
8.2.0	EMS-3589	The EMS administration tool options `-pwdfile` and `-ssl_pwdfile` were previously not documented. These options are now described in the *TIBCO Enterprise Message Service User's Guide*.
Issues Closed in Release 8.1.0		
8.1.0	EMS-5771	Fixed an issue that could prevent the automatic removal of a dynamic topic if a parent topic had at least one consumer with pending messages. A manifestation of this defect could be the accumulation of temporary topics on a server, if those temporary topics originated from a routed server, for instance in the context of fast pace request/reply messages.
8.1.0	EMS-5770	Previously, the Central Administration server sometimes failed to notify users that the EMS server required a restart after certain configuration changes were deployed. This has been fixed.
8.1.0	EMS-5766	Fixed an error that caused Central Administration to reject configuration changes to existing multicast channels. The Central Administration server now accepts the changes. After deployment, the EMS server requires a restart before the changes take effect.
8.1.0	EMS-5764	Fixed an error that could cause memory loss on startup.
8.1.0	EMS-5762	Fixed formatting issues in Central Administration.
8.1.0	EMS-5760	Fixed an error that sometimes caused Central Administration to report that it had created a queue ACL even though the desired topic ACL was correctly created.
8.1.0	EMS-5751	Fixed an error that caused a small memory loss in JSON-configured servers.
8.1.0	EMS-5718	In Central Administration, entries for the Processors to Bind to Network IO field are now validated to ensure only integer values are accepted.

Closed in Release	Key	Summary
8.1.0	EMS-5692	Given a fault-tolerant server pair A1 and A2, in which a global topic G is bridged to a local topic L, and with a route to server B which also defines a global topic G, in the event that server A1 fails and A2 becomes active, a consumer on local topic L will stop receiving messages from publishers connecting to server B and publishing on topic G. This has been fixed.
8.1.0	EMS-5682	Previously, the EMS Schema Export Tool did not function with JSON-configured EMS servers on zLinux, Solaris, AIX, and HP platforms. This has been fixed.
8.1.0	EMS-5677	Fixed a defect in EMS CA where unchecking the boxes for the route or factory SSL "Verify Host" and "Verify Hostname" had no effect. By default these are enabled even when the boxes are not checked. If you want to disable them and the boxes are not checked, you must check them and then uncheck them for the disable to take effect.
8.1.0	EMS-5647	Fixed an issue that would cause an unexpected `txcommit` trace when messages sent with a delivery time became available.
8.1.0	EMS-5645	Fixed an issue that could cause the server to exit abruptly when deleting the connection ID `1`.
8.1.0	EMS-5644	Fixed an issue with JSON-configured EMS servers that caused an ACL creation to fail with a no memory error.
8.1.0	EMS-5627	Fixed an issue that could cause unacknowledged messages sent to dynamic destinations to be recovered after a server restart if parent destinations had an expiration override property set. The server would no longer expire those messages. Note that only one message per dynamic destination would be affected by this defect.
8.1.0	EMS-5511	Fixed an issue that could cause memory loss when making JSON-based configuration changes.
8.1.0	EMS-5505	Fixed a syntax error that prevented Import Tranport and Export Transport from working correctly in Central Administration.

8.1.0	EMS-5499	Fixed an issue related with the use of synchronous file stores that would cause the following error message to be printed in stdout:
		`DEBUG: Insufficient buffer`
		and this error message in the log/console:
		`SEVERE ERROR: Failed writing message to '<file name>': I/O error or out of disk space.`
		With some EMS Server releases and when a transacted session is used to send messages, this error message could be seen as well:
		`ERROR: Abandoning transaction record due to IO failure`

8.1.0	EMS-5498	Fixed an issue that could cause a standby EMS server to crash if the server was shutdown or killed while it was in the process of activating.

Closed in Release	Key	Summary
8.1.0	EMS-5494	Previously, if an external user was added to a group on a JSON-configured server, the user would be created as well. This behavior differs from that of servers configured using `.conf` files, and has been corrected. Now servers running in either configuration mode will no longer add external users to the configuration.
8.1.0	EMS-5422	Fixed an issue that could prevent the EMS server from starting when processor IDs were specified (`processor_ids` in the server configuration and/or `processor_id` in a store configuration), if the given processor ID fell outside the range of online processors on this machine.
		The new behavior is that if the ID of a processor that is offline (or that falls outside of the list of online processors) is specified, the server fails at the time it tries to bind a network or storage thread to that given processor. The server still fails while parsing the configuration if an incorrect value is specified, such as a non numeric or negative value.
8.1.0	EMS-5420	Fixed an issue that would cause the `show durable(s)` command to show a durable as being online even though that durable consumer was closed, as long as its session and connection were still opened.
8.1.0	EMS-5411	Fixed an issue that caused memory loss when resetting multicast statistics.
8.1.0	EMS-5410	Fixed an issue that caused memory loss when removing a consumer from a topic that imports from SmartSockets.
8.1.0	EMS-5408	Fixed an issue that sometimes caused a memory loss in the EMS server when the admin tool was used to remove an imported or exported transport.
8.1.0	EMS-5406	Fixed an error that could cause a memory loss when using the `showacl user` *username* command in the administration tool.
8.1.0	EMS-5402	Previously, the EMS server would sometimes print a "slow clock tick" message if the recovery of store files took longer than 10 seconds. This has been fixed.
8.1.0	EMS-5400	Fixed an error that could cause clients with a connection timeout set to double-close the socket if the EMS server accepted but then quickly closed the connection.
8.1.0	EMS-5398	Fixed an issue that could cause the "Consumers" count of the `show topic` *topic-name* command to be incorrect in the presence of offline durable subscribers.
8.1.0	EMS-5388	Fixed an issue where the EMS .NET client ignored the selector provided to the QueueBrowser constructor.
8.1.0	EMS-5379	The JAAS module examples did not correctly allow Active Directory group back-link searches. This is now supported with the prebuilt JAAS modules.

Closed in Release	Key	Summary
8.1.0	EMS-5378	Fixed an issue that would cause the server to accept more client connections than were authorized by the `max_connections` parameter. This problem occurred when the server had clients using fault tolerant URLs and was either restarted or experienced a failover.
8.1.0	EMS-5376	Fixed an issue that could cause an EMS client to throw an exception when recovering an expired message for a closed consumer.
8.1.0	EMS-5369	Fixed an issue that would cause UFO Shared consumers to become Unshared consumers after their connection is recovered.
8.1.0	EMS-5364	Fixed an issue in the EMS Java client that could cause a NullPointerException in `Tibjms.getAsBytes()` when processing a message that was not received from a consumer session. Examples of this are messages created by `Tibjms.createFromBytes()` or from a QueueBrowser.
8.1.0	EMS-5361	EMS now properly rejects a subscription name that is null or is an empty string when creating a shared (durable or non-durable) consumer.
8.1.0	EMS-5349	Fixed an issue that could cause a C application using a tibemsUFOConnectionFactory to crash or use incorrect values for the message selector and/or client ID strings, after it reconnects to an active EMS Server.
8.1.0	EMS-5344	Previously, if a user was added to a group they already belonged to on a JSON-configured server, the server would report an error. It now ignores the add request.
8.1.0	EMS-5333	Fixed an issue that could prevent creation of a route (producing the error: "Implicit route to [<route name>] already exists") in a multi-hop routing setup and when a route between other servers was previously deleted.
8.1.0	EMS-5297	When using mstores, messages consumed from the `$sys.undelivered` queue in a transaction that didn't cleanup before server shutdown could reappear upon server restart.
8.1.0	EMS-4982	Previously, Central Administration did not prevent users from modifying existing durable consumers. This has been fixed. If a durable is defined, Central Administration now informs users attempting to modify it that the existing durable must be deleted and recreated with the desired settings.
8.1.0	EMS-4162	Fixed an issue that would cause the number "Total Acked" in the `show consumers full` command output for topic consumers to be higher than the "Total Sent". This problem occurred after messages were discarded due to the destination's `maxMsgs` or `maxBytes` properties.
8.1.0	EMS-2632	Fixed an issue that sometimes caused an `ERROR: stores file 'stores.conf'' does not exist` message when using the EMS Schema Export tool on Windows systems.

Closed in Release	Key	Summary
8.1.0	EMS-2488	Fixed an error that could cause memory loss when the `routes.conf` file was misconfigured.
Issues Closed in Release 8.0.0		
8.0.0	EMS-5295	When a store definition contains unknown properties, the server now reports the configuration error. It will fail to start if the `startup_abort_list` contains `CONFIG_ERRORS`.
8.0.0	EMS-5287	Messages received as part of an XA transaction (by an application using fault tolerant URLs) may not be redelivered if a communication error occurs while ending this transaction (for instance if the EMS server is not reachable or is performing a failover), and yet be committed as part of the next transaction.
8.0.0	EMS-5256	Fixed an error that caused memory leaks during a server deployment through Central Administration.
8.0.0	EMS-5247	Fixed an error that caused memory leaks when Central Administration was used to add, modify, or revoke ACLs.
8.0.0	EMS-5216	Fixed an issue that could cause the EMS Server to crash when EMS Java clients (version 6.0+) called `QueueBrowser.close()` after the queue had been administratively deleted.
8.0.0	EMS-5214	Fixed an issue which caused server to discard too many messages on a queue using mstore-based store when queue had `overflowPolicy=discardOld`.
8.0.0	EMS-5186	Fixed a memory leak that occurred when a route disconnected, if the route was previously in a stalled state.
8.0.0	EMS-5179	Fixed an issue that would prevent messages on the system undelivered queue `$sys.undelivered` to be browsed or consumed after a server restart, if those messages originally belonged to a queue with `maxRedelivery` property and a store of type `mstore`, and those messages were moved to `$sys.undelivered` after the `maxRedelivery` limit was reached.
8.0.0	EMS-4887	Fixed an error that prevented the `ssl_dh_size` parameter from taking effect when set using Central Administration.
8.0.0	EMS-4750	Previously, a Central Administration validation error was generated if spaces were added between trace options when specifying `log_trace` or `console_trace` settings. This has been fixed to allow leading and trailing white space in a comma separated list.
8.0.0	EMS-4681	Previously, the working copy of the EMS server JSON configuration file did not always match the configuration file that would be deployed. Certain fields, such as obfuscated passwords, were not transformed until deployment. This has been corrected so that the displayed working copy always shows exactly what will be sent to the server upon deployment.

Closed in Release	Key	Summary
8.0.0	EMS-4665	Fixed an error that caused the Central Administration server to open the jetty connector at a random port.
8.0.0	EMS-4662	Previously, the Central Administration page showed a redeploy option for failed deployments. This has been fixed. Only successful previous deployments can be redeployed.
8.0.0	EMS-4655	The help option for the EMS server has been updated to include descriptions for `-config` with JSON files, `-secondary`, and `-forceStart`.
8.0.0	EMS-4654	Fixed an error that caused all EMS servers configured with JSON configuration files to log that they were "Configured as fault tolerant primary", regardless of the actual settings.
8.0.0	EMS-4579	Fixed an error that could cause the EMS server to start successfully even with an invalid stores configuration.
8.0.0	EMS-4360	Fixed an error that could cause an EMS standby server to fail when mstores were configured and certain administrative commands were issued to that server, including `set server track_message_ids`.
8.0.0	EMS-4327	Fixed an error that sometimes caused message loss when messages were rolled back to a destination with `overflowPolicy=discardOld` and mstores configured.
8.0.0	EMS-3897	Fixed an error that caused an application's connection to be unusable—and show as stopped in the tibemsadmin tool—if the client library tried to connect to a non-EMS server process that was incorrectly part of the FT URL list.
8.0.0	EMS-2651	Fixed an issue that would cause the C API call `tibemsBytesMsg_GetBytes()` to return `TIBEMS_INVALID_ARGUMENT` if the received message had an empty body. It now returns `TIBEMS_OK` and a byte size of zero.

Known Issues

The table lists known issues in the listed version of TIBCO Enterprise Message Service

Key	Summary/Workaround
EMS-7694	**Summary**: Installing EMS 8.5.0 using `zypper install` on version 15 of Novell SUSE Linux Enterprise Server on x86-64 may result in the installer signaling that the corresponding EMS packages are not signed, which is expected. **Workaround**: Since the EMS RPM packages are not signed, ignore the corresponding warning.
EMS-7685	**Summary**: Beginning with release 6.1, the `ldap_tls_cacert_dir` server property has been subject to a check upon server startup that verifies if the configured directory exists. On Windows only, this check always fails, preventing a server configured with that property from ever starting up. **Workaround**: Chain the files contained within the target directory into the existing file used for the `ldap_tls_cacert_file` property and remove the `ldap_tls_cacert_dir` property from the server configuration file.
EMS-7242	**Summary**: If changes were made directly to the EMS server configuration using the Administration Tool or the Admin API and the corresponding configuration in Central Administration was not refreshed, deploying subsequent changes made in Central Administration will fail. The Actions drop-down menu on the server deployment log page can then be used to investigate and remedy the situation. However, reverting the configuration through the Revert icon and then selecting the Overwrite option of the Actions drop-down menu results in an error message in Central Administration. **Workaround**: Reverting a configuration leaves it unlocked. Lock it again before selecting the Overwrite option of the Actions drop-down menu.
EMS-7189	**Summary**: On certain 7.x releases of Red Hat Enterprise Linux (and all Linux distributions that are materially equivalent) older than 7.3, an EMS C client using an FT URL with its second or further member containing either `localhost` or the hostname of the local machine may not be able to connect to the EMS server on the local machine because of an issue with the hostname resolution. **Workaround**: In place of `localhost` or the local machine's hostname, provide the C client with the local machine's IP address or its loopback IP address.
EMS-7061	**Summary**: Expired messages that have been moved to the `$sys.undelivered` queue from a topic and that are consumed from `$sys.undelivered` right before stopping the server may be redelivered when the server is restarted. **Workaround**: None.
EMS-6401	**Summary**: Starting with EMS 8.2.2, the presence of a valid CRL file that is empty of revoked certificates in the `ssl_crl_path` directory will trigger a warning. Such a warning encountered at startup time will cause the EMS server to abort if the `startup_abort_list` holds the `SSL` condition. **Workaround**: If the `startup_abort_list` holds the SSL condition, make sure that no valid CRL file that is empty of revoked certificates is placed in the `ssl_crl_path` directory.

Key	Summary/Workaround
EMS-6082	**Summary**: The version of the `libeay32.dll` shared library that is included with EMS is built to optionally support FIPS. This has a side-effect of preventing its relocation in a process address space during run time. **Workaround**:If your Windows application fails to start due to a relocation error, try these workarounds: Relink your application with the `/FIXED` flag. Relink your application with static libraries.
EMS-3702	**Summary**: The JMSHeader fields of messages imported from SmartSockets cannot be modified. **Workaround**: None.

EMS-3088	**Summary**: The EMS server does not load OCI drivers (used with the OracleRAC database server).
	
	 Workaround: In order to load the OCI libraries, specify the driver location using the `module_path` parameter in the `tibemsd.conf`. For example:
	 `module_path=/usr/Oracle11gClient/linux24gl23/x86/oci32`
	
	 Note that TIBCO SmartSockets users also use the `module_path` parameter to dynamically load the SmartSockets library files. In order to define both OCI and SmartSockets library locations, separators should follow the same conventions used to specify `PATH`. On Unix platforms separate paths using a colon (:). On Windows platforms, use a semicolon. For example:
	 `module_path=c:\tibco\ss\bin\i86_w32;c:\Oracle11gClient\oci32`

EMS-2192	**Summary**: SmartSockets transports are not supported on HP-UX. **Workaround**: None.
EMS-2156	**Summary**: During recovery, a server using database stores receives the following error, and startup fails: `ORA-00904: "THIS_"."TXNREC_STORE_ID": invalid identifier` This is related to a known issue with Hibernate. **Workaround**: Restart the server. On restart, the `tibemsd` recovers correctly, with no messages lost.