

TIBCO Enterprise Message Service™

Release Notes

*Software Release 8.3.0
April 2016*

Important Information

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

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

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

TIBCO, Two-Second Advantage, The Power of Now, TIB, Information Bus, TIBCO Enterprise Message Service, TIBCO Rendezvous, TIBCO Enterprise, TIBCO SmartSockets, TIBCO ActiveMatrix BusinessWorks, and TIBCO Hawk are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

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

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

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

Copyright © 1997-2016 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

Contents

Preface	v
Related Documentation	vi
TIBCO Enterprise Message Service Documentation	vi
Other TIBCO Product Documentation	vi
Third Party Documentation	vii
Typographical Conventions	viii
Connecting with TIBCO Resources	xi
How to Join TIBCOCommunity	xi
How to Access TIBCO Documentation	xi
How to Contact TIBCO Support	xi
Release Notes	1
New Features	2
Release 8.3	2
Release 8.2	4
Release 8.1	6
Release 8.0	8
Changes in Functionality	11
Release 8.3	11
Release 8.2	12
Release 8.1	13
Release 8.0	13
Deprecated and Removed Features	15
Deprecated Features	15
Removed Features	16
Platform Support	18
Migration and Compatibility	20
Migrating from Release 8.X	20
Migrating from Release 7.X	21
Reverting to an Earlier Release from Release 8.2 or 8.3	21
Closed Issues	22
Known Issues	40

Preface

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.

Topics

- [Related Documentation, page vi](#)
- [Typographical Conventions, page viii](#)
- [Connecting with TIBCO Resources, page xi](#)

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Enterprise Message Service Documentation

The following documents form the TIBCO Enterprise Message Service documentation set:

- *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 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>.

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i> <i>ENV_NAME</i> <i>EMS_HOME</i>	<p>TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments.</p> <p>An installation environment consists of the following properties:</p> <ul style="list-style-type: none">• Name Identifies the installation environment. This name is referenced in documentation as <i>ENV_NAME</i>. If you specify a custom environment name, on Microsoft Windows the name becomes a component of the path to the product shortcut in the Windows Start > All Programs menu.• Path The folder into which the product is installed. This folder is referenced in documentation as <i>TIBCO_HOME</i>. The value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco. <p>TIBCO Enterprise Message Service installs into a directory within <i>TIBCO_HOME</i>. This directory is referenced in documentation as <i>EMS_HOME</i>. The value of <i>EMS_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\tibco\ems\8.3.</p>
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use MyCommand to start the foo process.</p>
bold code font	<p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none">• In procedures, to indicate what a user types. For example: Type admin.• In large code samples, to indicate the parts of the sample that are of particular interest.• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [enable disable]

Table 1 General Typographical Conventions (Cont'd)




Convention	Use
<i>italic font</i>	<p>Italic font is used in the following ways:</p> <ul style="list-style-type: none"> • To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>. • To introduce new terms. For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: <code>MyCommand PathName</code>
Key combinations	<p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: <code>Ctrl+C</code>.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: <code>Esc, Ctrl+Q</code>.</p>
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
[]	<p>An optional item in a command or code syntax.</p> <p>For example:</p> <p><code>MyCommand [optional_parameter] required_parameter</code></p>
	<p>A logical OR that separates multiple items of which only one may be chosen.</p> <p>For example, you can select only one of the following parameters:</p> <p><code>MyCommand para1 param2 param3</code></p>

Table 2 Syntax Typographical Conventions

Convention	Use
{ }	<p>A logical group of items in a command. Other syntax notations may appear within each logical group.</p> <p>For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4.</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command requires two parameters. The first parameter can be either param1 or param2 and the second can be either param3 or param4:</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command can accept either two or three parameters. The first parameter must be param1. You can optionally include param2 as the second parameter. And the last parameter is either param3 or param4.</p> <pre>MyCommand param1 [param2] {param3 param4}</pre>

Connecting with TIBCO Resources

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcommunity.com>.

How to Access TIBCO Documentation

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

<https://docs.tibco.com/products/tibco-enterprise-message-service>

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site at <https://docs.tibco.com>.

How to Contact TIBCO Support

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

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

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

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

<https://support.tibco.com>

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

Release Notes

This document includes release notes for TIBCO Enterprise Message Service, Software Release 8.3.0.

Check the TIBCO Product Support web site at <https://support.tibco.com> for product information that was not available at release time. Entry to this site requires a username and password. If you do not have a username, you can request for one. You must have a valid maintenance or support contract to use this site.

Topics

- [New Features, page 2](#)
- [Changes in Functionality, page 11](#)
- [Deprecated and Removed Features, page 15](#)
- [Migration and Compatibility, page 20](#)
- [Closed Issues, page 22](#)
- [Known Issues, page 40](#)

New Features

This section lists features added since the last major (8.0.0) release of this product.

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 chapter on Working 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 Chapter 10, 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 on Configuring 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 .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 section on Monitoring Servers 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 this product.

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.

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** As of this release, most commands in the administration tool are now unavailable when a server using a JSON configuration file is not in the active state. In such a situation, the only commands 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 8.3.0 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
32-bit Server and Client Libraries	<p>The 32-bit versions of the server executables and client libraries are deprecated.</p> <p>Support for the 32-bit server executables will be removed in the next release.</p> <p>Support for the 32-bit client libraries will be removed in a future release.</p>	8.3.0
Multicast	<p>This release deprecates the support of the multicast feature of EMS based on the <code>tibemsmcd</code> (multicast daemon). Support for this feature will be removed in the next release.</p>	8.3.0
SSL Communication	<p>The following cipher suites are deprecated. These cipher suites are considered to be less than secure and may be dropped in a future release of EMS:</p> <ul style="list-style-type: none"> • <code>SSL_RSA_WITH_DES_CBC_SHA</code> • <code>SSL_DHE_DSS_WITH_DES_CBC_SHA</code> • <code>SSL_DHE_RSA_WITH_DES_CBC_SHA</code> 	8.3.0
Support of Microsoft Visual Studio 2010	<p>This release deprecates the support of Microsoft Visual Studio 2010 (also known as VC10). Visual Studio 2010 will no longer be supported in a future release. C and .NET developers on Windows platforms will need to upgrade to Visual Studio 2012 (also known as VC11).</p>	8.2.2

Affected Component	Description	Deprecated in Release
Admin API	<p>The Admin API restart methods are deprecated and will stop working in a future release. The deprecated methods are:</p> <ul style="list-style-type: none"> • Java: <code>TibjmsAdmin.restart()</code> • .NET: <code>Admin.Restart()</code> 	8.2.2
Entrust SSL Libraries	Support for Entrust libraries with EMS clients for Java using SSL will be dropped after this release.	8.2.0
JAAS and JACI	This release deprecates the <code>jaas_classpath</code> and <code>jaci_classpath</code> parameters. Users should migrate to the new <code>security_classpath</code> parameter.	8.1.0
Client Libraries	The <code>TopicInfo.getDurableCount</code> Java admin method and equivalent C and .NET methods are deprecated. Instead, use <code>TopicInfo.getDurableSubscriptionCount</code> , and equivalent methods in C and .NET.	8.0.0

Removed Features

Affected Component	Description	Deprecated in Release	Removed in Release
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 10 (x86) platforms.	N/A	8.2.2

Affected Component	Description	Deprecated in Release	Removed in Release
SSL Communication	<p>The following cipher suites are no longer supported in the EMS Java client:</p> <ul style="list-style-type: none"> 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 <p>The following cipher suites are no longer supported in the EMS .NET client:</p> <ul style="list-style-type: none"> EXP-RC2-CBC-MD5 EXP-RC4-MD5 RC4-MD5 	N/A	8.2.2
Stores	<p>The 32- and 64-bit <code>tibemsdb5revert</code> executables are no longer included in the TIBCO Enterprise Message Service packages.</p> <p>This tool was used to revert EMS store files from software release 5.x to a format compatible with 4.x.</p>	N/A	8.2.0
C Client API	<p>Because the C API does not support character conversion, <code>tibemsBytesMsg_ReadUTF</code> and <code>tibemsBytesMsg_WriteUTF</code> have been removed.</p>	N/A	8.1.0

Affected Component	Description	Deprecated in Release	Removed in Release
TIBCO Hawk	<p>The <code>com.tibco.tibjms.admin.hawk</code> 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 <code>com.tibco.tibjms.admin.hawk</code> package is built into the installation.</p> <ul style="list-style-type: none">With Hawk 4.9 (that ships with TRA 5.8.0), the microagent is still embedded in <code>hawk/4.9/lib/tibjmsadmin.jar</code>, which may cause a conflict with <code>ems/7.0/lib/tibjmsadmin.jar</code> 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: <code>hawk/5.0/plugin/ems/hawkemshma.jar</code> 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
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	

Platform	Status	As of Release	Notes
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 8.3.0 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 8.3.0 transports for TIBCO FTL or Rendezvous, you must configure the `module_path` parameter to include the path to the EMS libraries before the FTL and Rendezvous libraries. This is the case when working with transports for:

- TIBCO FTL 4.3.0 and FTL 4.2.0
- TIBCO Rendezvous 8.4.4 and earlier

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

Migrating from Release 8.X

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 or 8.3

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 closed issues in version 8.3.0 of TIBCO Enterprise Message Service.

Closed in Release	Key	Summary
Issues Closed in Release 8.3.0		
8.3.0	EMS-6644	<p>In previous releases, the SmartSockets <code>T_MSG_FT_CHAR</code> type was converted to an EMS Byte type while the EMS Character type was converted to the SmartSockets <code>T_MSG_FT_INT2</code> type.</p> <p>With EMS 8.3, the latter doesn't change but a SmartSockets <code>T_MSG_FT_CHAR</code> type is now converted to an EMS Character type.</p>
8.3.0	EMS-6625 EMS-6617	Some applications may populate the <code>JMSReplyTo</code> 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 <code>tibemsMsg_MakeWriteable</code> .
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.
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 <code>trace</code> property on a routed queue.

Closed in Release	Key	Summary
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	<p>Only on AIX, the <code>libldap</code> 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:</p> <pre>0509-150 Dependent module (...)/liblber.so could not be loaded.</pre> <p>This has been fixed.</p>
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	<p>Using the <code>/</code> UNIX path separator in the <code>logfile</code> server property when <code>logfile_max_size</code> was also set would cause log file rotation to fail on Windows. This has been fixed.</p> <p>On UNIX systems, a warning is now logged when the <code>logfile</code> property contains the <code>\</code> Windows path separator.</p>
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 <code>console_trace</code> or <code>log_trace</code> 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 <code>TIBEMS_NO_MEMORY</code> 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.

Closed in Release	Key	Summary
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 <code>sender_name</code> or <code>sender_name_enforced</code> 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.
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 <code>localhost</code> (or no hostname), now rely on the name resolution of <code>localhost</code> 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 <code>logfile</code> properties of two EMS servers in a fault tolerant pairs were different, the <code>tibemsconf2json</code> tool would not take that into account. This has been fixed: the resulting JSON configuration file will now include the corresponding <code>secondary_logfile</code> 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 <i>server name</i> > Monitor > Stores > <i>store name</i> > 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 <code>incoming_topic</code> property was specified with a topic name that was not present in the configuration.

Closed in Release	Key	Summary
8.3.0	EMS-6199	If a consumer on a routed queue received a message and exited before acknowledging it, the <code>JMSXDeliveryCount</code> 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.
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 <code>Bad resource name</code> 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 <code>sender_name</code> or <code>sender_name_enforced</code> 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.

Closed in Release	Key	Summary
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 <i>queue-name@server-name</i> and <i>server-name</i> 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 (<code>\$sys.meta</code> , <code>\$sys.nonfailsafe</code> or <code>\$sys.failsafe</code>) was not configured, but its file name (<code>meta.db</code> , <code>async-msgs.db</code> or <code>sync-msgs.db</code>) 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 <code>ldap_operation_timeout</code> 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 <code>SSL handshake failed: ret=-1, reason=<unknown></code> at startup if its peer was not yet present.

Issues Closed in Release 8.2.2

8.2.2	EMS-6236	<p>Fixed an issue that could cause the following error message:</p> <p>Failed writing message to '<i>store-file-name</i>': I/O error or out of disk space.</p> <p>This could happen when moving compressed persistent queue messages on synchronous stores to the undelivered queue.</p>
8.2.2	EMS-6226	The EMS server crashed when it accepted too many client connections concurrently. This has been fixed.

Closed in Release	Key	Summary
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 <code>logfile</code> or <code>ssl_crl_path</code> 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 <code>tibemsadmin</code> 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 <code>wait-for-peer</code> state, the <code>State.get()</code> admin API call returned 11 instead of returning the <code>State.SERVER_STATE_WAIT_FOR_PEER</code> 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 <code>max_connections</code> 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 <code>tibemsConnection_Close()</code> , 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 <code>JMSDeliveryTime</code> for messages imported from Rendezvous or SmartSockets was not set and defaulted to zero instead of its correct value. This has been fixed.

Closed in Release	Key	Summary
8.2.0	EMS-5873	Fixed an error that caused the <code>tibemsconf2json</code> tool to truncate the target destination of a bridge if that destination had the word <code>selector</code> 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: <code>authentication failed</code> .
8.2.0	EMS-5851	When creating a new store in Central Administration, the store type for <code>mstore</code> was mistakenly displayed as <code>Multiple store</code> in the type dropdown box. This has been corrected to show the store type as <code>mstore</code> .
8.2.0	EMS-5849	Fixed an issue that could potentially delay clients' clock synchronization when the server parameter <code>clock_sync_interval</code> was specified and clients' connections were closed or lost. The following warnings could also be produced: <ul style="list-style-type: none"> • 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 <code>JMS_TIBCO_SENDER</code> 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 <i>TIBCO Enterprise Message Service User's Guide</i> 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 <code>Server_A+Server_A+Server_B</code> and <code>Server_A</code> was down, the UFO client never connected to <code>Server_B</code> . In this scenario, the unshared state client will now connect to <code>Server_B</code> .

Closed in Release	Key	Summary
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.
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	<p>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.</p> <p>When the routed queue consumer was started on a proxy server, the following warning message would appear on the home server:</p> <p>WARNING: Routed Queue 'myQueue' is not a home Queue</p>
8.2.0	EMS-5806	<p>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:</p> <ul style="list-style-type: none"> • Closing a durable with unacknowledged messages • Re-opening (and later closing) the durable using a different session • Unsubscribing the durable subscription <p>Crashes could occur in different places, including but not limited to consumers statistic gathering, dynamic destination cleanup, session recover, and so on.</p>

Closed in Release	Key	Summary
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 (<code>clock_sync_interval</code> defined in the server), or when messages had a <code>JMSExpiration</code> set.
8.2.0	EMS-5776	The following parameters have been added to the output of the <code>show config</code> command in the administration tool: <ul style="list-style-type: none"> • <code>processor_ids</code> • <code>network_thread_count</code> • <code>selector_logical_operator_limit</code> • <code>max_msg_print_size</code> • <code>max_msg_field_print_size</code>
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 <code>ssl_trusted_list</code> or a <code>ssl_issuer_list</code> in the EMS JSON configuration file, when added through the administration tool.
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 <code>tibemsd.sh</code> and <code>tibemsd64.sh</code> did not have executable permissions and could not be invoked from other directories. In EMS 8.2.0, <code>tibemsd.sh</code> and <code>tibemsd64.sh</code> are now installed with executable permissions and can be invoked from any directory.

Closed in Release	Key	Summary
8.2.0	EMS-3589	The EMS administration tool options <code>-pwdfile</code> and <code>-ssl_pwdfile</code> were previously not documented. These options are now described in the <i>TIBCO Enterprise Message Service User's Guide</i> .

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.
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.

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

Closed in Release	Key	Summary
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.
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	<p>Fixed an issue that could prevent the EMS server from starting when processor IDs were specified (<code>processor_ids</code> in the server configuration and/or <code>processor_id</code> in a store configuration), if the given processor ID fell outside the range of online processors on this machine.</p> <p>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.</p>
8.1.0	EMS-5420	Fixed an issue that would cause the <code>show durable(s)</code> 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 <code>showacl user <i>username</i></code> 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.

Closed in Release	Key	Summary
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 <i>topic-name</i> 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.
8.1.0	EMS-5378	Fixed an issue that would cause the server to accept more client connections than were authorized by the <code>max_connections</code> 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 <code>NullPointerException</code> in <code>Tibjms.getAsBytes()</code> when processing a message that was not received from a consumer session. Examples of this are messages created by <code>Tibjms.createFromBytes()</code> or from a <code>QueueBrowser</code> .
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 <code>tibemsUFOConnectionFactory</code> to crash or use incorrect values for the message selector and/or client ID strings, after it reconnects to an active EMS Server.

Closed in Release	Key	Summary
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 <code>\$sys.undelivered</code> 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 <code>show consumers full</code> command output for topic consumers to be higher than the "Total Sent". This problem occurred after messages were discarded due to the destination's <code>maxMsgs</code> or <code>maxBytes</code> properties.
8.1.0	EMS-2632	Fixed an issue that sometimes caused an <code>ERROR: stores file 'stores.conf' does not exist</code> message when using the EMS Schema Export tool on Windows systems.
8.1.0	EMS-2488	Fixed an error that could cause memory loss when the <code>routes.conf</code> 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 <code>startup_abort_list</code> contains <code>CONFIG_ERRORS</code> .
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.

Closed in Release	Key	Summary
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 <code>QueueBrowser.close()</code> 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 <code>mstore</code> -based store when queue had <code>overflowPolicy=discardOld</code> .
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 <code>\$sys.undelivered</code> to be browsed or consumed after a server restart, if those messages originally belonged to a queue with <code>maxRedelivery</code> property and a store of type <code>mstore</code> , and those messages were moved to <code>\$sys.undelivered</code> after the <code>maxRedelivery</code> limit was reached.
8.0.0	EMS-4887	Fixed an error that prevented the <code>ssl_dh_size</code> 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 <code>log_trace</code> or <code>console_trace</code> 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.
8.0.0	EMS-4665	Fixed an error that caused the Central Administration server to open the jetty connector at a random port.

Closed in Release	Key	Summary
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 <code>-config</code> with JSON files, <code>-secondary</code> , and <code>-forceStart</code> .
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 <code>set server track_message_ids</code> .
8.0.0	EMS-4327	Fixed an error that sometimes caused message loss when messages were rolled back to a destination with <code>overflowPolicy=discardOld</code> 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 <code>tibemsBytesMsg_GetBytes()</code> to return <code>TIBEMS_INVALID_ARGUMENT</code> if the received message had an empty body. It now returns <code>TIBEMS_OK</code> and a byte size of zero.

Known Issues

The table lists known issues in version 8.3.0 of TIBCO Enterprise Message Service.

Key	Summary/Workaround
EMS-6665	<p>Summary Beginning with release 8.3, the addition or the deletion of stores of type <code>file</code> made through Central Administration do not require restarting the standby server of a fault-tolerant pair of EMS servers anymore. Restarting the active server is enough. However, modifications made to the properties of an existing store of type <code>file</code> are still not effective in the standby server until it is restarted.</p> <p>Workaround After modifying the properties of an existing store of type <code>file</code> in Central Administration, follow the instruction provided in the Central Administration documentation to restart both the active and the standby servers.</p>
EMS-6401	<p>Summary Starting with EMS 8.2.2, the presence of a valid CRL file that is empty of revoked certificates in the <code>ssl_crl_path</code> directory will trigger a warning. Such a warning encountered at startup time will cause the EMS server to abort if the <code>startup_abort_list</code> holds the SSL condition.</p> <p>Workaround If the <code>startup_abort_list</code> holds the SSL condition, make sure that no valid CRL file that is empty of revoked certificates is placed in the <code>ssl_crl_path</code> directory.</p>
EMS-6217	<p>Summary VMware Tools prevents the installer from registering the TIBCO EMS .NET client assemblies in the Windows GAC.</p> <p>This issue impacts only installations of EMS on Windows virtual machines that are running on VMWare products.</p> <p>Workaround Uninstall VMware Tools. Install TIBCO EMS. Reinstall VMware Tools.</p>
EMS-6082	<p>Summary The version of the <code>libeay32.dll</code> 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.</p> <p>Workaround If your Windows application fails to start due to a relocation error, try these workarounds:</p> <ol style="list-style-type: none"> 1. Relink your application with the <code>/FIXED</code> flag. 2. Relink your application with static libraries.

Key	Summary/Workaround
—	<p>Summary Installing EMS 7.0 in a <i>TIBCO_HOME</i> environment where EMS 8.0 has already been installed is not supported.</p> <p>Workaround If required, install EMS 7.0 in a <i>TIBCO_HOME</i> environment separate from that of EMS 8.0.</p>
EMS-4105	<p>Summary When configuring SSL on z/Linux, authentication fails if a PEM format certificate with P7 suffix is specified to the IBM JRE.</p> <p>Workaround Modify the certificate to be in DER format with a P7 suffix.</p>
EMS-3702	<p>Summary The JMSHeader fields of messages imported from SmartSockets cannot be modified.</p> <p>Workaround None.</p>
EMS-3162	<p>Summary Using both multicast and the SmartSockets gateway at the same time is not supported.</p> <p>Workaround None.</p>
—	<p>Summary The EMS server does not load OCI drivers (used with the OracleRAC database server).</p> <p>Workaround In order to load the OCI libraries, specify the driver location using the <code>module_path</code> parameter in the <code>tibemsd.conf</code>. For example:</p> <pre>module_path=/rv/tools/tibjms/Oracle11gClient/linux24gl23/x86/oci32</pre> <p>Note that TIBCO SmartSockets users also use the <code>module_path</code> 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 <code>PATH</code>. On Unix platforms separate paths using a colon (:). On Windows platforms, use a semicolon. For example:</p> <pre>module_path= c:\tibco\ss\bin\i86_w32;c:\Oracle11gClient\oci32</pre>
EMS-3049	<p>Summary When running EMS in FIPS compliant mode, DSA certificates cannot be used.</p> <p>Workaround Use RSA certificates when running EMS in FIPS compliant mode.</p>
EMS-2192	<p>Summary The SmartSockets gateway is not supported for the 64-bit EMS server on the HP-UX 11i v3 platform.</p> <p>Workaround None.</p>

Key	Summary/Workaround
EMS-2156	<p>Summary During recovery, a server using database stores receives the following error, and startup fails:</p> <p>ORA-00904: "THIS_". "TXNREC_STORE_ID": invalid identifier</p> <p>This is related to a known issue with Hibernate.</p> <p>Workaround Restart the server. On restart, the <code>tibemsd</code> recovers correctly, with no messages lost.</p>
EMS-384 1-22ZRNM	<p>Summary JSSE cannot read PKCS12 certificates generated by some versions of OpenSSL.</p> <p>Workaround Import the certificate into a web browser; then export the certificate to a new file with extension <code>.p12</code> (not <code>.pfx</code>).</p>