TIBCO Enterprise Message Service™

Release Notes

Software Release 5.1.5
April 2010

>>>TIBCO®
The Power of Now®

Contents

Release Notes

This document includes release notes for TIBCO Enterprise Message Service, Software Release 5.1.5.

Check the TIBCO Product Support web site at http://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 one, you can request one. You must have a valid maintenance or support contract to use this site.

Topics

New Features

This section lists features and the release they were added to this product.

Reference # and Release	Feature
New Features in Release 5.1	
5.1	**Unshared State Failover**
	This release adds a new connection factory type, `TibjmsUFOConnectionFactory`, which supports unshared state failover. When an EMS client creates a server connection using a unshared state connection factory, the client can include servers that do not share state in the server URL list.
	`TibjmsUFOConnectionFactory` is available only in Java.
5.1	**Microsoft DTC**
	TIBCO Enterprise Message Service now supports Microsoft Distributed Transaction Coordinator (MS DTC). The MS DTC allows .NET clients to participate in distributed transactions.
5.1	**FIPS 140-2 Compliance**
	The EMS server and clients can now be enabled to operate in compliance with the Federal Information Processing Standard (FIPS), Publication 140-2. FIPS 140-2 compliance is available on Windows, Linux, and Solaris 10 (x86) platforms.
5.1	**Database Support**
	This release adds support for the Oracle Real Application Clusters (RAC) 10g database, using the Oracle Instant Client 11g, which is a light-weight JDBC OCI driver.
5.1	**Increased Character Limits for Store Parameters**
	The configuration parameters that specify store settings have increased character limits. Specifically, `dbstore_driver_url` now supports long URL names and file-based stores support long file names.
	Additionally, the `dbstore_driver_username` and `dbstore_driver_password` parameters now support *username* and *password* values of up to 255 characters.

Reference # and Release	Feature
5.1	**New C APIs**
	Three new functions have been added to the C client library:
	• `tibemsConnection_IsDisconnected`
	• `tibemsConnectionFactory_SetUserName`
	• `tibemsConnectionFactory_SetPassword`
	See the *TIBCO Enterprise Message Service C & COBOL Reference* for more information about these functions.
5.1 1-1P4HUE	The EMS server now maintains an exact redelivery count for queues with a positive `prefetch` count. The server and clients mark messages as redelivered (and increase the message delivery count) only after explicitly generated rollback events. These events include calls to `session.recover` or `session.rollback`, or closed connections. In the event of an application crash, the exact count is not guaranteed.
5.1 1-9F6S2B	Performance has been increased, significantly decreasing the length of time the `TibjmsAdmin.getTopics()` and `TibjmsAdmin.getQueues()` methods take to retrieve large numbers of topics or queues from the server.
5.1 1-96Z5PL	This release includes performance improvements in the server when queue consumers disconnect, and there are pending acknowledgments in XA transactions left in "prepare" mode.
5.1 1-8VM9HJ	This release introduces improved EMS server memory and process usage when acknowledging `PERSISTENT` messages for topics with multiple durable subscribers (or subscribers with fault tolerant URLs).
5.1	EMS now uses `/dev/poll`, instead of `select()`, on Solaris.

Reference # and Release	Feature
New Features in Release 5.0	
5.0	**Extensible Security**
	The extensible security feature allows you to write your own authentication and permissions modules using the Java Authentication and Authorization Service (JAAS) for authentication modules, and the Java Access Control Interface (JACI) for permissions modules. These modules run in a Java virtual machine (JVM) in the EMS server, and authenticate users and authorize them to perform actions such as publish and subscribe operations.
	For more information on extensible security, see the related chapter in the *TIBCO Enterprise Message Service User's Guide*.
5.0	**Store Messages in Multiple Stores, Now Including Relational Databases**
	The store feature allows you to store messages to one or more files or database instances, filtering them by destination. Stores are defined in the `stores.conf` configuration file, and associated with a destination using the `store` destination property.
	With this feature, you can configure your messaging application to store messages in different locations for each application, or to create separate backup files for related destinations. Because stores are configured in the server, they are transparent to clients.
	This feature is described in further in the *TIBCO Enterprise Message Service User's Guide*.
5.0	**Administration API Added to C**
	The C client administration API provides functions to monitor an EMS system. The C administration library is a subset of the Java and .NET administration APIs. The library provides the ability to query the EMS application, and get information about existing clients, servers, and destinations.
	For more information about the C administration API, see the *TIBCO Enterprise Message Service C & COBOL Reference*.

Reference # and Release	Feature
5.0	**Multicast** Multicast is a messaging model that allows the EMS server to send messages to multiple consumers simultaneously by broadcasting them over an existing network. Instead of delivering a copy of the message to each individual subscriber over TCP, the EMS server broadcasts the message over Pragmatic General Multicast (PGM). A daemon running on the machine with the subscribed EMS client receives the multicast message and delivers it to the message consumer. See the *TIBCO Enterprise Message Service User's Guide* for more information about multicast.
5.0	Added support for the .NET Compact Framework API on Windows 32 VC8 and Windows 64.
1-6ZS6Q8 5.0	Added a method to the client library that can be used to direct client tracing to a file, when tracing is enabled. Note that tracing must be enabled in the server in order for this method to have effect. See `tibems_SetTraceFile`, `Tibjms.setTraceFile` and `Tibems.SetTraceFile` in C, Java and .NET reference documentation for more information, and the *TIBCO Enterprise Message Service User's Guide* for more information about client tracing.
1-6QTXJZ 5.0	Added the ability to reset a user's password by executing a script within the administration tool. For more information, see the description of the `-script` administration tool option in the *TIBCO Enterprise Message Service User's Guide*.
1-7X62OH 5.0	Added statistics gathering for the `$sys.undelivered` queue. In previous releases, messages moved to and consumed from the undelivered message queue would not be recorded with statistics for the original destination. Beginning with EMS 5.0, undelivered message statistics are included in statistics generated for the original destination.
1-8JY74F 5.0	Added the ability to set a different queue limit policy for each Rendezvous transport. The queue limit policy limits the number of pending messages in Rendezvous queues. For more information, see the section on configuring transports for Rendezvous in the *TIBCO Enterprise Message Service User's Guide*.
1-8AHWB3 5.0	Added a function to get a certificate store from the connection factory object. A new .NET API, `GetCertificateStore`, was added to the connection factory.

Reference # and Release	Feature
1-82BSYW 5.0	Improved the compact command to decrease the amount of time needed to compact a file-based store.
1-6Z6PGV 5.0	Added the ability to delete temporary destinations with a matching wildcard pattern using the administration tool or API. The `delete` command all now allows you to delete temporary destinations that do not have any consumers.
New Features in Release 4.4	
1-6QKMMF 4.4	Added support for the following Windows platforms: • VC8 32-bit (win32/x86/vc8) • VC8 64-bit (win64/x86) Note that the .Net Compact Framework API is supported only for Windows 32 VC7. It is NOT supported for Windows 32 VC8 and Windows 64.
1-7TCHJT 4.4	Added support for the Mac OSX x86 (darwin/x86) platform.
1-7BXZB6 4.4	Improved server starting time when a significant number of bridges are configured.
1-77D64V 4.4	Added sample Java clients, `tibjmsPerfMaster.java` and `tibjmsPerfSlave.java` to test the performance of various EMS options. The master client spawns EMS producers and the slave client spawns EMS consumers. The master client instructs the slave to set consumer options, and the slave reports the performance results of the options back to the master.
1-752VGF 4.4	Improved performance of .NET client when writing int's, char's, longs, short's to the network stream.
1-72BASJ 4.4	Improved error reporting in response to a fault-tolerant URL containing a bad username/password for the client.
1-71X0LT 4.4	Improved event reporting for I/O failures that occur when EMS is writing to the shared state system.
1-71SMF6 4.4	Added a prefetch property to topics so that users can define the prefetch value for topics.

Reference # and Release	Feature
1-71A5S5 4.4	Makefiles for the C samples updated to provide the ability to specify which libraries to link with 64-bit or 32-bit.
1-712UCO 4.4	EMS server now checks its routing configuration in the `routes.conf` file against what is held in the `meta.db` at startup time. If these settings do not agree, the EMS server will not start.
1-6ZBY9B 4.4	Added detailed error code to identify whether authentication failure is the result of a firewall authentication error or the wrong EMS server URL or port.
1-6XYEP9 4.4	Updated the `info` command to display the count of consumers and producers.
1-6WLJDF 4.4	.NET API updated with the `tibemsd.getConnectionActiveURL()` method to return the URL of the EMS server for a connection.
1-6T98HW 4.4	Extended support for the destination properties, `overflowPolicy` and `maxmsgs`, to include topics.
1-6RFDJS 4.4	Updated the `showacl user` command and added `GetPermissions()` methods to the Java and .NET client admin APIs to show the effective permissions for an admin or user, either direct or inherited from its groups.
1-6QM4TJ 4.4	The EMS plugin has been upgraded so that the microagent's name remains "com.tibco.tibjms.admin.hawk.HawkController(or Listener)" and its display name is "HawkListener[or Controller](tcp://host:port)". In order to preserve backwards compatibility, you can uncomment the `-server_in_agent_name` argument in the "arguments" section in the `tibjmsadmin.hma` file to force the plugin back to the old behavior.
1-5ZD341 4.4	Upgraded EMS server to send heartbeat messages to clients so that clients can detect server disconnect faster than the TCP layer.
1-5QEFK6 4.4	Added the ability to mangle the password associated with the `ldap_credential` configuration in the `tibemsd.conf` file.
1-5KTZSF 4.4	Added server ldap parameters to the Java and .NET admin APIs.

Reference # and Release	Feature
1-54IQ6J 4.4	Added an option for producers to not wait for the server's confirmation to NON_PERSISTENT messages when server's authorization is enabled.
1-52U7KH 4.4	Improved error reporting in the C client by adding a `getLastErrorString()` function that returns the error string out of the last error reply from the server.
1-4YSZ0A 4.4	Added a `show consumers` command and extended the Java and .NET admin APIs to output the number of pending messages to detect slow non-durable subscribers.
1-4V8BG5 4.4	Updated C API so that objects are reference counted. Applications must now explicitly call the `tibems_close()` function to deallocate the memory used by the EMS internal global data structures.
1-4I205H 1-2CEHA6 4.4	Updated the EMS server to avoid a 'size limit reached' error when the ldap cache is enabled.
1-4BIQ2I 4.4	Updated the EMS server to allow the same clientID and durable subscriber name to be used when connecting to the backup server after a fault-tolerant switchover.
1-49NB94 4.4	Updated the C API to include a `tibemsMsg_PrintToBuffer` function to print message to a buffer.
1-3RCXZC 4.4	Updated the `getTopic()` and `getQueue()` methods in the Java and .NET admin APIs to return only select destinations, such as "notemp," "static," or "dynamic."
1-36RB11 4.4	Added the connection id to the `CONNECT` and `CONNECT_ERROR` tracing messages.
1-22U3FM 4.4	Added monitoring messages to flag when messages are expired by the server.
1-1HAVPS 1-18SXI1 4.4	Extended .NET API to support SSL and message compression.

Reference # and Release	Feature
1-1EPAR2 4.4	Updated the Administration Tool and client APIs with the ability to set a timeout value on establishing a connection to the server.
New Features in Release 4.3	
4.3	SSL key renegotiation is deprecated.
1-3U1SFY 1-18SXHC 4.3	Two new properties configure queue behavior: • `maxmsgs` limits the number of unconsumed messages in a queue. • `overflowPolicy` offers a more flexible response when a queue overflows its `maxmsgs` or `maxbytes` limits.
4.3	Release 4.3 supports OpenVMS on Alpha and Itanium hardware.
1-6MVDM9 1-6WAW41 4.3	Administrative users can now authenticate against an LDAP.
1-6N8KKT 4.3	Java and C clients now can connect to an EMS server using an SSL connection through a client-side firewall. New connection factory entry points get and set the SSLProxy parameters.
1-6PEDUE 4.3	Java Rendezvous transport objects in sending programs can now specify a sequence of EMS servers, and connect to the first available server.
1-6E42VP 4.3	EMS under z/OS now supports SSL.
New Features in Release 4.2	
1-18SXGF 4.2	SSL for Authentication Only Some applications require strong or encrypted authentication, but do not require message encryption. The configuration parameter `ssl_auth_only` specifies SSL only during the authentication phase; subsequent messages do not incur the overhead of encryption.

Reference # and Release	Feature
4.2	SSL Enhancements • Support for new SSL vendor `j3se-default` • Support for additional cipher suites in Java
1-UV9FQ 4.2	Dynamic Bridges Administrators can now create and destroy bridges without restarting the server; see the `tibemsadmin` commands `create bridge` and `delete bridge`. New administrator permissions `view-bridge` and `change-bridge` regulate this feature.
1-1IL54R 4.2	.NET Administration API .NET programs can use this new API to administer the EMS server. This API is functionally identical to the Java Administration API.
1-19TJZE 4.2	Permission to Use Durables The ACL can now distinguish between permission to use an existing durable, and permission to create and configure a durable.
1-1WABEB 4.2	Enhanced Durable Information Enhanced output from the administration tool command `show durable` (and the equivalent calls in the administration APIs). This information now includes the number of messages delivered to a durable.
1-22I3WV 4.2	Bridge Permission Bridges have automatic permission to send to their target destinations.
1-1SEXX9 4.2	Enhanced Client Trace Administrators can now set the `client_trace` parameter in the main configuration file (previous releases could set this parameter only using the administration tool).
1-3TYWXA 4.2	Undelivered SmartSockets GMD Messages A new parameter, `preserve_gmd`, in `transports.conf` determines the behavior of the EMS server when it has exported a GMD message to SmartSockets, and SmartSockets cannot deliver that message. When SmartSockets returns the undelivered message, EMS can either preserve it in the EMS undelivered message queue, or discard it.

Reference # and Release	Feature
1-1FL6C9 4.2	Message Memory Usage In the Administration APIs, a new method of the `ServerInfo` class lets you get the message memory usage information from a server.
1-266AOV 4.2	EMS adds support for these AES cipher suites: • `TLS_RSA_WITH_AES_128_CBC_SHA` • `TLS_RSA_WITH_AES_256_CBC_SHA`
1-2HIA2R 1-457Q07 4.2	Added a new EMS installation packages for Linux 2.4 with glibc 2.3.
1-47KDZR 4.2	Enhanced server treatment of invalid file handles during operation.
1-4LXGKA 4.2	Added a new Hawk microagent parameter, `-encryptedPassword`, which accepts a password encpyted with the Hawk utility program `tibhawkpassword`. This feature lets you store an encrypted password in the `.hma` file.
1-4LXTSL 4.2	Connected fault-tolerant servers now detect a situation in which both are simultaneously active.
1-25XVVT 4.2	Enhanced server performance. Restarting the server is now much faster in situations where many messages exist that have been acknowledged by some durable subscribers, but not by all subscribers.
1-43EH3P 4.2	Added new syntax for specifying cipher suites in the EMS server and in C clients. The `/` qualifier replaces the existing cipher list, overriding existing `!` prefixes (which disable their ciphers).
1-519X6I 4.2	Enhanced administration tools and APIs to extract `user_auth` settings.
1-3NP0B9 4.2	Enhanced output from the Hawk plug-in.
4.2	New documentation is available for the C & COBOL APIs. This new book, *C & COBOL Reference*, replaces the old *C Reference* book.

Reference # and Release	Feature
New Features in Release 4.1.0	
1-23WU8U 4.1.0	Enhanced output from the Hawk admin plug-in method `getServerInfo`. This method now includes more information about the server, such as message memory usage and process id.
1-1VYVEJ 4.1.0	New `-timeout` parameter for Hawk MicroAgent.
1-1HAVPH 4.1.0	.NET Compact Framework lets handheld devices use EMS.
1-1WPNN9 4.1.0	Importing messages from SmartSockets now maps more SmartSockets headers to EMS properties.
1-1N8BT9 4.1.0	This release introduces a new book to the documentation set—*TIBCO Enterprise Message Service .NET API Reference.*
1-1G0DUF 4.1.0	.NET API now supports `TopicRequestor` and `QueueRequestor` objects.
1-1IQ7YE 4.1.0	C programs can now access EMS administered objects in external LDAP name servers. New entry points in the C API and Java API enable this enhancement. In earlier releases, only Java programs could access these objects through JNDI calls. Now both Java and C programs can access these objects in the same name server.
1-1JH35P 4.1.0	Performance improvements when the server accumulates a large number of pending messages for which the `Expiration` header is non-zero.
1-1BRT8A 4.1.0	Administrators can now promote an active-passive route to an active-active route without restarting either server.
1-18T9EP 4.1.0	The utility `emsntsrg` registers `tibemsd` as a Windows service, so it can start automatically.
1-1VCFKF 4.1.0	Process ID (PID) now identifies the server process in logs and traces, as well as the `show server` command (in the administration tool and administration APIs).

Reference # and Release	Feature
1-1IKYUA 4.1.0	Servers remap route URLs periodically when they cannot connect.
1-1O4S1X 4.1.0	Enhanced log file rotation. Rotation failure disables rotation. Any successful attempt to manually rotate the logs (using the administration tool or APIs) re-enables rotation.
New Features in Release 4.0.0	
1-1LFSWF 4.0.0	.NET API enhancements: • This release adds event handler and delegate classes for message consumers. This architecture is a better fit with .NET programming style, and using it can yield performance improvements over the older callback architecture. For backward compatibility, this release continues to support the callback architecture. • This release adds enumerated constants to denote delivery mode and acknowledgement mode. New overloaded methods accept these `enum` values instead of the older constants. Using them improves reliability of application programs through stronger type checking at compile time. For backward compatibility, this release continues to support the older methods and constants.
1-1L4033 4.0.0	Performance improvements when Java clients access compressed messages. Apparent improvement depends on message size.
1-1L3HOR 4.0.0	When translating imported messages from TIBCO Rendezvous, fields of type `TIBRVMSG_XML` now translate to byte arrays in EMS messages. (In earlier releases, they erroneously translated to strings.) For backward compatibility with client programs that process XML fields as strings, the configuration parameter `tibrv_xml_import_as_string` restores the previous behavior). Nonetheless, we strongly recommend coding all new application programs to process XML fields as byte arrays.
1-18QC8G 4.0.0	TIBCO SmartSockets interoperability. A new external transport type let you create a bridge between `tibemsd` and SmartSockets RTserver.

Reference # and Release	Feature
1-18T9EV 4.0.0	Four new connection factory properties configure attempts to connect and reconnect to servers: • `connect_attempt_count` • `connect_attempt_delay` • `reconnect_attempt_count` • `reconnect_attempt_delay`
1-1DADL2 4.0.0	Four new server parameters configure heartbeat and timeout intervals for connections to clients and to other servers: • `client_heartbeat` • `client_connection_timeout` • `server_heartbeat` • `server_connection_timeout`
1-14SG4O 4.0.0	Documentation for server error messages now appears in an appendix to *TIBCO Enterprise Message Service User's Guide.*
1-18QCAM 4.0.0	Administrators can compact server database files to reclaim unused disk storage.
1-18QC87 4.0.0	Routing among servers allows both 1-hop and multi-hop zones, for greater control.
1-18T9FH 4.0.0	You can configure connection factories for load balancing among servers.
1-UV9FA 4.0.0	`trace_client_host` is a new parameter in `tibemsd.conf`, which determines the way in which trace statements identify client hosts.
1-VLPEQ 4.0.0	Servers support certificate revocation lists (CRLs). Two new parameters in `tibems.conf` affect this feature—`ssl_crl_path` and `ssl_crl_update_interval`. A new administration tool command, `updatecrl`, immediately updates a server's CRL.
1-19TJYZ 4.0.0	A new configuration file, `durables.conf`, allows static configuration of durable subscribers. This feature helps avoid message loss when starting a large system.

Reference # and Release	Feature
1-18QC9M 4.0.0	A new property of destinations lets administrators override the `expiration` property of messages.
1-1859TE 4.0.0	`startup_abort_list` is a new parameter in `tibems.conf`. Administrators can specify a variety of conditions that cause the server to exit during its initialization sequence.
1-18T9F2 4.0.0	TIBCO EMS servers can use encrypted connections to LDAP servers. Several new configuration file parameters pertain to this feature: • `ldap_conn_type` • `ldap_tls_cacert_file` • `ldap_tls_cacert_dir` • `ldap_tls_ciphers` • `ldap_tls_rand_file` • `ldap_tls_cert_file` • `ldap_tls_key_file`

Compatibility With Previous Versions

Release 5.1.5

There are no compatibility issues in this release.

Release 5.1.4

There are no compatibility issues in this release.

Release 5.1.3

Many of the files included in the TIBCO Enterprise Message Service release 5.1.3 are identical to the files included in the 5.1.0 release. These files include most libraries, header files, and binary files. The updated files, which include the 5.1.3 version number, are:

- `tibemsd`
- `tibemsd64`
- `tibemsd.jar`

Release 5.1.2

Many of the files included in the TIBCO Enterprise Message Service release 5.1.2 are identical to the files included in the 5.1.0 release. These files include most libraries, header files, and binary files. The updated files, which include the 5.1.2 version number, are:

- `tibemsd`
- `tibemsd64`
- `tibemsd.jar`

Additionally, the Mac OSX platforms include new versions of the EMS C client libraries that are compatible with the OpenSSL 0.9.8k libraries.

SmartSockets Libraries are Dynamically Loaded

In previous releases, the EMS server statically loaded SmartSockets libraries. The EMS server now dynamically loads the libraries. If your system uses SmartSockets, you must enter a new parameter into the main `tibemsd.conf` configuration file in order to let the EMS server know where the SmartSockets libraries are located:

```
module_path = SmartSockets-shared-library-directory
```

where *SmartSockets-shared-library-directory* is the absolute path to the directory containing the SmartSockets library files. For example:

```
module_path = c:\tibco\ss\bin\i86_w32
```

Please see the SmartSockets documentation to locate the directory where the appropriate shared libraries are installed.

UNIX   These UNIX libraries must be included in the specified directory:

- `librtutil_o`
- `librtedm_o`
- `librtsm_o`
- `librtconn_o`
- `librtmsg_o`
- `librtsubj_o`
- `librtipcl_o`
- `librtipc_o`

Note that a 32-bit EMS server requires the 32-bit SmartSockets libraries, whereas a 64-bit EMS server requires the 64-bit SmartSockets libraries.

Windows   These Windows DLLs must be included in the specified directory:

32-bit Server	64-bit Server
• `tutil50.dll`	• `tutil64.dll`
• `tedm50.dll`	• `tedm64.dll`
• `tsm50.dll`	• `tsm64.dll`
• `tconn50.dll`	• `tconn64.dll`
• `tmsg50.dll`	• `tmsg64.dll`
• `tsubj50.dll`	• `tsubj64.dll`
• `tipcl50.dll`	• `tipcl64.dll`
• `tipc50.dll`	• `tipc64.dll`

Release 5.1.1

Many of the files included in the TIBCO Enterprise Message Service release 5.1.1 are identical to the files included in the 5.1.0 release. These files include all libraries, header files, and most binary files. The updated files, which include the 5.1.1 version number, are:

- `tibemsd`

- `tibemsd64`

- `tibemsd.jar`

Additionally, the 5.1.1 release includes one new file for the .NET platform:

- `TIBCO.EMS.UFO.dll`

Release 5.1

Updating the Database Schema

The 5.1 release of TIBCO Enterprise Message Service includes 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` option to apply these changes to your database store implementation.

This update is necessary for applications using database stores with the MS DTC feature in .NET clients, and to implement fixes related to issue numbers 1-93NXTO and 1-9G6CPD.

MS DTC Transactions

When using the MS DTC feature, leaving a prepared MS DTC transaction at the EMS server can cause incorrect transaction recovery in 5.0 (or older) EMS clients. Resolve the prepared MS DTC transaction by either committing it or rolling it back before starting older client applications.

Release 5.0

Reverting From Release 5.0 to an Earlier Release

When you upgrade from EMS 4.x to 5.0 (or later), `tibesmd` makes some changes to your store files that allow them to function with the multiple store feature. These changes are incompatible with 4.x, and will prevent a 4.x server from starting if you try to roll back to an older version of EMS.

The `tibemsdb5revert` tool is included with EMS in order to remove the changes made to those files. The tool takes one argument, which is the directory where the store files are located. By default, the files are in the `tibemsd` working directory. You must give the absolute path to the directory. For example:

```
> tibemsdb5revert /filesystem/datastore
```

Run `tibemsdb5revert` after reverting to 4.x, but before restarting the EMS server.

The tool only makes changes to default store files, `sync-msgs.db`, `async-msgs.db`, and `meta.db`. If you have altered the default configuration to change these file names, `tibemsdb5revert` fails.

Release 4.4

When upgrading from EMS 4.3 to 4.4 (or later) on OpenVMS platforms, EMS client executables that were linked with the EMS 4.3 dynamic libraries (shareable images) must be relinked to the new libraries after EMS 4.4 has been installed with its associated third party libraries. The third party libraries are part of the full installation of EMS.

Release 4

All clients created with any TIBCO Enterprise Message Service 4.x.x release are compatible with all other TIBCO Enterprise Message Service Release 3.x.x clients and servers. New features in Release 4, such as one-hop routing, and SmartSockets bridges, may not work with clients and servers from older releases. If you are not using any new Release 4 features, clients and servers from all 3.x.x releases should be completely compatible.

You may run the 4.x.x administration tool with the 3.x.x release of the server. However, we discourage running the 3.x.x administration tool with release 4.x.x servers, since new features of release 4 may not work properly with earlier versions of the administration tool. It is a good idea to use the same version of the administration tool as the latest release of the product.

Changes in Functionality

This section lists changes to product functionality and the release when the change was introduced.

Release 5.1.5

Open SSL

TIBCO Enterprise Message Service now operates with OpenSSL version 0.9.8n.

Linux Package Name

On linux platforms, the installer package suffix has changed to `linux26gl23_x86.tar.gz`.

Release 5.1.4

Reference # and Release	Functionality
Release 5.1.4	
5.1.4	TIBCO Enterprise Message Service now supports the Oracle 11g database server, using JDBC thin driver `ojdbc6.jar`.
5.1.4	TIBCO Enterprise Message Service now operates with OpenSSL version 0.9.8l.
5.1.4	If you use the EMS Java client library with Entrust as the SSL vendor, you must either: • Upgrade to Entrust version 7.2 (or later), or • Install unlimited strength policy files in the JVM.
5.1.4	In release 5.1.4 and later, TIBCO Enterprise Message Service Java client library is no longer supported on z/OS v1.6. z/OS v1.7 and later is supported.

Reference # and Release	Functionality

5.1.4	To integrate TIBCO EMS with application servers using SSL, you must now add the following JAR files, installed in *EMS_HOME*/`lib`, to the CLASSPATH:
	
	`slf4j-api-1.4.2.jar`
	`slf4j-simple-1.4.2.jar`
	
	Note that `tibcrypt.jar` must also be in the CLASSPATH.

Release 5.1.3

There are no changes in functionality in this release.

Release 5.1.2

Reference # and Release	Functionality
Release 5.1.2	
5.1.2	TIBCO Enterprise Message Service now operates with OpenSSL version 0.9.8k.

Release 5.1.1

Reference # and Release	Functionality
Release 5.1.1	
5.1.1	**Unshared State Failover in .NET Clients** This release adds support for unshared state failover in the .NET client. Previously, this feature was available only in Java clients.

Release 5.1

Reference # and Release	Functionality
Release 5.1	
5.1	The TIBCO Enterprise Message Service installation process now installs user data and a subset of sample EMS configuration files in a different directory. In previous releases, these files were installed in the `bin` directory within *EMS_HOME*. They are now located in a directory specified during the Configuration Directory step of the installation process.
	Consequently, you must now use this new directory in order to start the preconfigured `tibemsd`.
	For more information, see the section on starting the EMS server in the *TIBCO Enterprise Message Service User's Guide*.
5.1	Beginning with TIBCO Enterprise Message Service release 5.1, the `tibemsd` and `tibemsmcd` run as administrator on Windows systems, enabling multicast functionality and permitting the `tibemsd` to modify the configuration files.
5.1	During installation, `TIBCO.EMS.DLL` and `TIBCO.EMS.ADMIN.DLL` are installed in the global assembly cache. If an earlier release of TIBCO Enterprise Message Service is installed on the computer, the EMS 5.1 DLLs replace those of the earlier version.
	To reinstall the previous EMS release DLLs, you must first uninstall the 5.1 DLLs, then reinstall the DLLs from the previous release. Both of these tasks are accomplished using the Global Assembly Cache Tool (`Gacutil.exe`) provided by Microsoft. DLL files are located in the bin directory of the EMS installation.
5.1	TIBCO Enterprise Message Service now operates with OpenSSL version 0.9.8j.
	The EMS server now dynamically loads these libraries (as well as zlib) for easier security upgrades.

Release 5.0

Reference # and Release	Functionality
Release 5.0	
5.0	The TIBCO Enterprise Message Service installation directory structure has changed, following TIBCO standards. TIBCO Enterprise Message Service is now installed in a version-specific folder. Additionally, the `clients` folder has been removed, and its contents redistributed in a standard TIBCO format. For example: • `ems/clients/c/include` is now `ems/5.0/include` • `ems/clients/c/lib` is now `ems/5.0/lib` • `ems/clients/c/bin` is now `ems/5.0/bin` • `ems/clients/java` is now `ems/5.0/lib` • `ems/cs` is now `ems/5.0/bin`
5.0	Beginning with EMS release 5.0, the server's default error recovery policy is to stop immediately when a corrupt record error is detected. In releases earlier than 5.0, the EMS server attempted to retrieve the active state regardless of any errors that were found. Some errors caused the server to proceed with the recovery while discarding the bad record or ignoring objects of the type associated with the error. Other errors caused the server to abort startup completely. The `-forceStart` command line option causes the `tibemsd` to delete corrupted messages in the store files, allowing the server to start even if it encounters errors. Using this option causes dataloss, and it is important to backup store files before using `-forceStart`.

Reference # and Release	Functionality
5.0	In the following instances, the message receivers attempting to acknowledge a message may throw the exception "Attempt to acknowledge message(s) not valid for this consumer":
	1. Attempting to acknowledge a message that's expired on the server side will cause this exception on the client.
	2. Sending an acknowledgement after a fault tolerant switch.
	The error occurs when the message no longer exists in the server. Prior to TIBCO Enterprise Message Service release 5.0, these situations would not have caused an error. The server would accept the acknowledgement even if the message no longer existed.
1-84GQMP 5.0	Increased the permitted line length in the `tibemsd.conf` configuration file from 256 characters to 256,000 characters. This prevents the EMS server from throwing a configuration error when parameter values exceed 256 characters.
1-7XLG9P 5.0	A destination's expiration property is now only applied to an imported TIBCO Rendezvous message when that message has a `JMSTimestamp` in the JMSHeaders sub-message. This now applies to RV messages that are imported to a destination that is bridged to a second destination.
1-8M8B3Y 5.0	Only a dynamic destination that is the direct source of a bridge is excluded from automatic clean-up. Dynamic destinations that serve as the source of a bridge due to inheritance are included in the automatic clean-up.
	In previous releases, all dynamically created destinations that were the sources of bridges, whether they were direct sources or inherited sources, were excluded from automatic clean-up.

Release 4.x

Reference # and Release	Functionality
Release 4.4.3	
4.4.3	This release of TIBCO Enterprise Message Service integrates with TIBCO Rendezvous® Software Release 8.1.0. Rendezvous users are strongly encouraged to install this upgrade release on all hardware.

Reference # and Release	Functionality
Release 4.4.2	
4.4.2	This release of TIBCO Enterprise Message Service integrates with TIBCO SmartSockets® Software Release 6.8.1. SmartSockets users are strongly encouraged to install this upgrade release on all hardware.
Release 4.4	
1-7D305C 4.4	In a fault-tolerant configuration, if an Administration Tool is connected to the backup server, it will be disconnected when a switchover occurs. This prevents a problem in previous EMS releases in which consumer clients might encounter incorrect exceptions, server leaks memory, and so on after switchover.
1-77OR6Z 4.4	The C client now detects and returns a `TIBEMS_ILLEGAL_STATE` when a session is used across threads.
1-5846ZX 4.4	In the C API, deleting a temporary topic from under a consumer now returns a TIBEMS_NOT_PERMITTED.
4.4	In order to eliminate the possibility of clients receiving duplicate messages from routed topics, the `consumed_msg_hold_time` flag has been replaced with a protocol change. In EMS 4.4, the consuming EMS server will hold routed messages until its local subscriber clients have fully consumed them and the consuming EMS server has verified that the producing EMS server has stored the acknowledgment. This guarantees that the producing EMS server will not send a duplicate routed message once the consuming EMS server has discarded its copy. Both the producing and consuming server must know about the new protocol before it will suppress duplicate routed messages. So, when upgrading to EMS 4.4, replace the producing EMS server first and then replace the consuming EMS server, or replace them both at the same time.
4.4	The behavior of the C function tibemsConnectionFactory_SetServerURL had changed to be consistent with its operation in Java and .NET. tibemsConnectionFactory_SetServerURL can be used to set the server URL for a connection factory only once. If the URL has previously been set for the connection factory, tibemsConnectionFactory_SetServerURL returns the status code TIBEMS_EXCEPTION.
4.4	Java Developer Kit (JDK) 1.3 is no longer supported. Use JDK 1.4 or later.

Reference # and Release	Functionality
Release 4.3	
1-6NAQOH 4.3	Two previously deprecated C API calls are now obsolete. For details, see Obsolete on page 37.
4.3	This releases supports Microsoft Visual C++ 7. EMS no longer supports Visual C++ 6.
4.3	Release 4.3 supports OpenVMS on Alpha and Itanium hardware.
Release 4.2	
1-18QCAC 4.2	Fixed a defect in which consumers could not acknowledge a message after the consumer is closed (contrary to the JMS specification). As part of this fix, server behavior has changed to comply with the JMS specification: When a client closes a consumer with an unacknowledged message, the server no longer reinstate that message back onto its queue. Although we strongly encourage you to correct application programs that mistakenly rely on the previous and erroneous behavior, we realize that in some situations such corrections might pose unanticipated difficulties. If your application relies on the server to reinstate unacknowledged messages after a consumer is closed, you can disable this defect correction by adding the following parameter setting to `tibemsd.conf`: `compliant_consumer_close = disabled` Otherwise, we strongly recommend leaving this parameter as `enabled` (the default value).
4.2	SSL If you use the EMS Java client library with Entrust as the SSL vendor, you must upgrade to Entrust 7.0 (or later). However, the vendor name constant remains `entrust61` (it has not changed).
4.2	In prior releases, the server could authenticate users based on information in the UNIX password file. This feature is deprecated as of release 4.2, and it is not available for Mac OS X.
1-318TAZ 4.2	Added a new C API entry point, `tibems_close()`, which Windows programs must call before unloading the EMS DLL.

Reference # and Release	Functionality
4.2	The C function `tibemsMsg_ByteSize` is deprecated. Use `tibemsMsg_GetByteSize` instead.
1-1YPX5Z 4.2	Name length limits are now enforced. Client ID names are limited to 255 bytes. Durable names are limited to 255 bytes.
1-33MCX5 4.2	Fixed the state name for FT standby state in the administration APIs (.NET and Java) `ServerInfo` class. Administration programs that used the old deprecated value (`SERVER_STANDBY`) require updating to the new value (`SERVER_FT_STANDBY`).
Release 4.1.0	
4.1.0	Support for SSL accelerators (such as Ingrian) is deprecated in release 4.1.0.
4.1.0	In release 4.1.0 and later, TIBCO Enterprise Message Service no longer supports Linux kernel 2.2, which is the basis of Red Hat Linux 6.2 (we continue to support Red Hat 7).
4.1.0	In release 4.1.0 and later, TIBCO Enterprise Message Service no longer supports AIX 4.3. We continue to support AIX 5.1.
Soon	Microsoft has announced that it will soon drop support for its Windows NT 4.0 Server operating system. At that time, we will also drop support for EMS on that platform.
4.1.0	The method `Session.Run` is obsolete.
4.1.0	In the Java API, the following methods of class `Tibjms` are deprecated: `getAllowCallbackInClose` and `setAllowCallbackInClose`. These items were misnamed; we recommend eliminating references to them from your code, replacing them with references to `getAllowCloseInCallback` and `setAllowCloseInCallback` (respectively). In the .NET API, the corresponding methods of class `Tibems` are deprecated: `GetAllowCallbackInClose` and `SetAllowCallbackInClose`. We recommend eliminating references to them from your code, replacing them with references to `GetAllowCloseInCallback` and `SetAllowCloseInCallback` (respectively).
1-1IQQZW 4.1.0	For new configuration details regarding JBoss 3.2.3, see *TIBCO Enterprise Message Service Application Integration Guide.*

Reference # and Release	Functionality
Release 4.0.0	
4.0.0	Release 4 begins a transition to a more comprehensive product—Enterprise Message Service (EMS). Accordingly, several items within the product have new names. For details, see Name Changes in Release 4 on page 28.
1-1L3HOR 4.0.0	When translating imported messages from TIBCO Rendezvous, fields of type TIBRVMSG_XML now translate to byte arrays in EMS messages. (In earlier releases, they erroneously translated to strings.) For backward compatibility with client programs that process XML fields as strings, the configuration parameter tibrv_xml_import_as_string restores the previous behavior). Nonetheless, we strongly recommend coding all new application programs to process XML fields as byte arrays.
1-18T9F9 4.0.0	When a fault-tolerant backup server becomes the new primary server, it first re-reads all of its configuration files, to ensure that it uses that latest parameter values.

Name Changes in Release 4

Release 4.0.0 signifies a transition from a JMS product to a more comprehensive message product—Enterprise Message Service (EMS). Accordingly, several items within the product have new names. Nonetheless, these name changes do not affect the validity of existing programs.

Executable Components

- tibemsd replaces tibjmsd.

- tibemsadmin replaces tibjmsadmin.

For backward compatibility, the installation includes two copies of each executable—one with the old name and one with the new name. The two copies have identical contents.

Configuration Files

- tibemsd.conf replaces tibjmsd.conf.

The server first attempts to locate an existing file named tibemsd.conf. If it does not find one, it then attempts to locate an existing file named tibjmsd.conf (for backward compatibility). If it does not find either file, it creates a new file named tibemsd.conf.

C API

- The header file `tibems.h` replaces `tibjms.h`.

  The installation includes a backward compatibility header file, `tibjms.h`, which declares the old-name functions.

- The library `libtibems` replaces `libtibjms`.

  For backward compatibility, the installation includes a *pass-through* library, `libtibjms`, with functions that call the corresponding functions in the new library, `libtibems`.

- Functions, types and other items that begin with the prefix `tibems` replace corresponding items that had the prefix `tibjms`.

- Functions that get and set message attributes omit the substring `JMS` from their names; for example, `tibemsMsg_GetTimestamp` replaces `tibjmsMsg_GetJMSTimestamp`.

- Constants that refer to TIBCO-specific message properties begin with the prefix `JMS_` have *not* changed; for example, `JMS_TIBCO_COMPRESS`.

- Example programs call new-name functions.

Java API

No names have changed within the Java API. The substring `JMS` remains as before.

C# API

The C# API already reflected this name change when we introduced it, so no names have changed in release 4.

Deprecated & Obsolete Features

This section describes deprecated features and lists equivalent features that accomplish the same result, if relevant. 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 equivalent feature.

Release 5.1

Deprecated

Failsafe

The `failsafe` parameter is deprecated. The same functionality can be achieved through the configuration of message stores and default store files.

Obsolete

Visual Studio 7

TIBCO Enterprise Message Service is no longer supported for Visual Studio 7.

Release 5.0

Obsolete

Configuration Parameters

- The `user_auth` configuration parameter no longer supports the `system` value.

- `subscribe_mode`—this parameter was used in the `transports.conf` file to determine how the EMS server passed TIBCO SmartSockets message subscriptions from EMS clients to RTserver, and whether subject filtering occured in the EMS server or in RTserver. All message filtering now occurs in RTserver.

Destination properties

The following destination properties, which were deprecated in release 4.4, are no longer supported:

Deprecated	Replacement
tibrv_import	import
tibrvcm_import	import
tibrv_export	export
tibrvcm_export	export

Platform Support

TIBCO Enterprise Message Service no longer supports HP Tru64 UNIX on Alpha or Linux on Itanium hardware.

C API

These API calls were deprecated in an earlier release, and are now obsolete. Delete these calls from your programs, substituting the appropriate replacement calls:

Obsolete	Replacement
tibemsMsg_PrintFile	tibemsMsg_PrintToBuffer
tibemsMsgField_PrintFile	tibemsMsgField_PrintToBuffer

Deprecated

Configuration File Parameters

- ssl_rand_file—this parameter from the tibemsd.conf file was used to locate a file containing random data, and could not be used in production environments. It is not needed for EMS applications running on current operating systems.

- store_crc, store_minimum, and store_truncate—these parameters are incompatible with the multiple store files feature. The same functionality can be achieved through store configuration in the stores.conf file.

- `tibrv_xml_import_as_string`—this parameter was used to revert the `tibemsd` to an earlier, incorrect behavior. When importing messages from Rendezvous, `tibemsd` translates XML fields to byte arrays. Releases earlier than 4.0 erroneously translated them to strings. This parameter caused tibemsd to revert to the earlier, incorrect behavior (strings).

C API

Deprecated	Replacement
Message Requestor API	
`tibemsTopicRequestor_Create` `tibemsQueueRequestor_Create`	`tibemsMsgRequestor_Create`
Connection Factory API	
API using connection factories to create connections, SSL connections, and XA connections. Note that connection factory types, with the exception of `generic`, are also deprecated.	

`tibemsTopicConnectionFactory_Create`
`Connection`
`tibemsQueueConnectionFactory_Create`
`Connection`
	`tibemsConnectionFactory_CreateConnection`

`tibemsConnectionFactory_Create`
`    ConnectionSSL`
`tibemsTopicConnectionFactory_Create`
`    ConnectionSSL`
`tibemsQueueConnectionFactory_Create`
`    ConnectionSSL`
	Use this call sequence:
	1. `tibemsConnectionFactory_SetServerURL`
	2. `tibemsConnectionFactory_SetSSLParams`
	3. `tibemsConnectionFactory_SetPkPassword`
	4. `tibemsConnectionFactory_CreateConnection`

`tibemsXAConnectionFactory_Create`
`    XAConnection`
`tibemsXATopicConnectionFactory_`
`    CreateXAConnection`
`tibemsXAQueueConnectionFactory_`
`    CreateXAConnection`
	`tibemsConnectionFactory_CreateXAConnection`

`tibemsXAConnectionFactory_Create`
`    XAConnectionSSL`
`tibemsXATopicConnectionFactory_`
`    CreateXAConnectionSSL`
`tibemsXAQueueConnectionFactory_`
`    CreateXAConnectionSSL`
	Use this call sequence:
	1. `tibemsConnectionFactory_SetServerURL`
	2. `tibemsConnectionFactory_SetSSLParams`
	3. `tibemsConnectionFactory_SetPkPassword`
	4. `tibemsConnectionFactory_CreateXA`
	`        Connection`

Deprecated	Replacement
`tibemsConnectionFactory_SetType`	There is no replacement for this function. When all connection factory types except `generic` are no longer supported, there will be no need for this function.
`tibemsConnectionFactory_GetType`	There is no replacement function. When all connection factory types except `generic` are no longer supported, there will be no need for this function.

Connection API

Connection API creating connections, SSL connections, and sessions.

Deprecated	Replacement
`tibemsTopicConnection_Create` `tibemsQueueConnection_Create`	`tibemsConnection_Create`
`tibemsTopicConnection_CreateSSL` `tibemsQueueConnection_CreateSSL`	`tibemsConnection_CreateSSL`

`tibemsTopicConnection_CreateTopic`
   `Session`
`tibemsQueueConnection_CreateQueue`
   `Session`
	`tibemsConnection_CreateSession`

Session API

Session API creating or deleting temporary destinations, consumers, producers, and so on.

Deprecated	Replacement
`tibemsTopicSession_CreateSubscriber` `tibemsQueueSession_CreateReceiver`	`tibemsSession_CreateConsumer`
`tibemsTopicSession_CreatePublisher` `tibemsQueueSession_CreateSender`	`tibemsSession_CreateProducer`
`tibemsTopicSession_CreateDurable` `Subscriber`	`tibemsSession_CreateDurableSubscriber`
`tibemsTopicSession_Unsubscribe`	`tibemsSession_Unsubscribe`
`tibemsTopicSession_CreateTemporary` `Topic`	`tibemsSession_CreateTemporaryTopic`
`tibemsTopicSession_DeleteTemporary` `Topic`	`tibemsSession_DeleteTemporaryTopic`
`tibemsQueueSession_CreateBrowser`	`tibemsSession_CreateBrowser`
`tibemsQueueSession_CreateTemporary` `Queue`	`tibemsSession_CreateTemporaryQueue`

Deprecated	Replacement
`tibemsQueueSession_DeleteTemporary Queue`	`tibemsSession_DeleteTemporaryQueue`
Miscellaneous Topic and Queue API	
`tibemsTopicPublisher_Publish` `tibemsQueueSender_Send`	`tibemsMsgProducer_Send`
`tibemsTopicPublisher_PublishEx` `tibemsQueueSender_SendEx`	`tibemsMsgProducer_SendEx`
`tibemsTopicPublisher_PublishToTopic` `tibemsQueueSender_SendToQueue`	`tibemsMsgProducer_SendToDestination`
`tibemsTopicPublisher_PublishToTopi cEx` `tibemsQueueSender_SendToQueueEx`	`tibemsMsgProducer_SendToDestinationEx`
`tibemsTopicPublisher_GetTopic` `tibemsQueueSender_GetQueue`	`tibemsMsgProducer_GetDestination`
`tibemsTopicSubscriber_GetTopic` `tibemsQueueReceiver_GetQueue`	`tibemsMsgConsumer_GetDestination`
`tibemsTopicSubscriber_GetNoLocal`	`tibemsMsgConsumer_GetNoLocal`
Deprecated Types	
`tibemsTopicConnection` `tibemsQueueConnection`	`tibemsConnection`
`tibemsTopicSession` `tibemsQueueSession`	`tibemsSession`
`tibemsTopicSubscriber` `tibemsQueueReceiver`	`tibemsMsgConsumer`
`tibemsTopicPublisher` `tibemsQueueSender`	`tibemsMsgProducer`
`tibemsTopicRequestor` `tibemsQueueRequestor`	`tibemsMsgRequestor`

`tibemsTopicConnectionFactory`
`tibemsQueueConnectionFactory`
`tibemsXAConnectionFactory`
`tibemsXATopicConnectionFactory`
`tibemsXAQueueConnectionFactory`
	`tibemsConnectionFactory`

Release 4.4

Obsolete

Platform Support

EMS is no longer supported on the linux24gl22/ia64 and linux24gl22/x86 platforms.

Rendezvous configuration parameters

The following Rendezvous configuration parameters have been replaced with the configuration parameters in the `transports.conf` file. These are obsolete and no longer supported (CR 1-4ABV41):

- `tibrv_bridge`

- `tibrv_service`

- `tibrv_network`

- `tibrv_topic_import_dm`

- `tibrv_queue_import_dm`

- `tibrv_export_headers`

- `tibrv_export_properties`

- `tibrvcm_enable`

- `tibrvcm_name`

- `tibrvcm_ledger`
- `tibrvcm_sync_ledger`
- `tibrvcm_request_old`
- `tibrvcm_default_ttl`

SSL accelerators

SSL accelerators are no longer supported.

Temporary Destination Compliance mode

Temporary Destination Compliance mode has been removed, as EMS is compliant by default (CR 1-1EDDR7 and 1-19RIU6).

C API

The type tibemsAnyConnectionFactory is has been removed from the C library.

The many connection factory types were condensed in to one general type. The new connection factory type is `tibemsConnectionFactory`.

Deprecated

C API

The following C functions are deprecated in release 4.4:

Deprecated	Replacement
`tibemsMsg_PrintFile`	`tibemsMsg_PrintToBuffer`
`tibemsMsgField_PrintFile`	`tibemsMsgField_PrintToBuffer`

Admin APIs

The `DestinationInfo` class `setDescription` and `getDescription` methods in the admin APIs are deprecated in release 4.4 (CR 1-3F6DOB).

Destination properties

The following destination properties are deprecated in release 4.4:

Deprecated	Replacement
tibrv_import	import
tibrvcm_import	import
tibrv_export	export
tibrvcm_export	export

Network connection failure parameters

The following network connection failure parameters are deprecated in release 4.4:

Deprecated	Replacement
client_heartbeat	client_heartbeat_server
client_connection_timeout	server_timeout_client_connection
server_heartbeat	server_heartbeat_server
server_connection_timeout	server_timeout_server_connection

Release 4.3

Obsolete

Two C API calls were deprecated in an earlier release, and are now obsolete. Delete these calls from your programs, substituting the appropriate replacement calls in Table 1.

Table 1   Obsolete C API Calls

Obsolete	Replacement
tibemsStreamMsg_GetBytes	tibemsStreamMsg_ReadBytes
tibemsStreamMsg_SetBytes	tibemsStreamMsg_WriteBytes

Obsolete

UNIX system password authentication is no longer supported.

Deprecated

The following features are deprecated in release 4.3:

- SSL key renegotiation
- The `system` value for the `user_auth` configuration parameter

Deprecated for Future Simplification

In release 5.0 we plan to condense the many subtypes of connection, connection factory, and session into a small number of general types. As a result many entry points will become obsolete, some function signatures will change, and the API as a whole will be simpler to understand and use.

The following C API entry points (and the corresponding COBOL entry points) are deprecated in release 4.3, and are scheduled for removal in release 5.0. Prepare to migrate to the more general calls and types. (This table indicates the expected migration path, based on the current design at the time of this writing. We provide this advance notice to help you plan ahead, but we do not guarantee that the final implementation will match it exactly.)

Deprecated	Equivalent Replacement
Connections	
Topic connection and queue connection simplify to the more general type—connection.	
`tibemsTopicConnection`	`tibemsConnection`
`tibemsQueueConnection`	`tibemsConnection`
Connection Factories	
Topic connection factory and queue connection factory simplify to the more general type—connection factory.	
`tibemsTopicConnectionFactory`	`tibemsConnectionFactory`
`tibemsQueueConnectionFactory`	`tibemsConnectionFactory`

Deprecated	Equivalent Replacement
XA Connection Factories	
All types of XA connection factory simplify to the more general type—connection factory.	

`tibemsXAConnectionFactory`
`tibemsXATopicConnectionFactory`
`tibemsXAQueueConnectionFactory`
	`tibemsConnectionFactory`

Connection Factory Type	
It is no longer necessary to distinguish the type of a connection factory.	
`tibemsConnectionFactoryType`	No longer needed.
`tibemsConnectionFactory_GetType`	No longer needed.
`tibemsConnectionFactory_SetType`	No longer needed.
Connection Factory: Create Connection	
The several types of connection factory each had several calls, to create various types of connections. These calls all condense into one general call.	
`tibemsTopicConnectionFactory_CreateConnection` `tibemsQueueConnectionFactory_CreateConnection`	`tibemsConnectionFactory_CreateConnection`

`tibemsConnectionFactory_CreateConnectionSSL`
`tibemsTopicConnectionFactory_CreateConnectionSSL`
`tibemsQueueConnectionFactory_CreateConnectionSSL`
	`tibemsConnectionFactory_CreateConnection`

`tibemsXATopicConnectionFactory_CreateXAConnection` `tibemsXAQueueConnectionFactory_CreateXAConnection`	`tibemsConnectionFactory_CreateConnection`

`tibemsXAConnectionFactory_CreateXAConnectionSSL`
`tibemsXATopicConnectionFactory_CreateXAConnectionSSL`
`tibemsXAQueueConnectionFactory_CreateXAConnectionSSL`
	`tibemsConnectionFactory_CreateConnection`

Deprecated	Equivalent Replacement
Connection: Create	
The several types of connection each had several create calls, to create various types of connections. In release 5.0 all of these calls are obsolete. To create a connection, set the appropriate parameters in a connection factory, and use `tibemsConnectionFactory_CreateConnection`.	
`tibemsConnection_Create`	`tibemsConnectionFactory_CreateConnection`
`tibemsTopicConnection_Create` `tibemsQueueConnection_Create`	`tibemsConnectionFactory_CreateConnection`
`tibemsConnection_CreateSSL` `tibemsQueueConnection_CreateSSL` `tibemsTopicConnection_CreateSSL`	`tibemsConnectionFactory_CreateConnection`
`tibemsXAConnection_Create` `tibemsXAConnection_CreateSSL`	`tibemsConnectionFactory_CreateConnection`
Connection: Create Session	
Two connection calls created two types of sessions. These calls condense into one general call.	
`tibemsQueueConnection_CreateQueueSession` `tibemsTopicConnection_CreateTopicSession`	`tibemsConnection_CreateSession`
Session	
The two types of session each had several calls. Each of these calls is replaced, either by an existing general call, or by a new call.	
`tibemsTopicSession_CreateSubscriber`	`tibemsSession_CreateSubscriber` (new call)
`tibemsTopicSession_CreateDurableSubscriber`	`tibemsSession_CreateDurableSubscriber`
`tibemsTopicSession_CreatePublisher`	`tibemsSession_CreatePublisher` (new call)
`tibemsTopicSession_Unsubscribe`	`tibemsSession_Unsubscribe`
`tibemsTopicSession_CreateTemporaryTopic`	`tibemsSession_CreateTemporaryTopic`
`tibemsTopicSession_DeleteTemporaryTopic`	`tibemsSession_DeleteTemporaryTopic`
`tibemsQueueSession_CreateBrowser`	`tibemsSession_CreateBrowser`

Deprecated	Equivalent Replacement
`tibemsQueueSession_CreateReceiver`	`tibemsSession_CreateReceiver` (new call)
`tibemsQueueSession_CreateSender`	`tibemsSession_CreateSender`
`tibemsQueueSession_CreateTemporaryQueue`	`tibemsSession_CreateTemporaryQueue`
`tibemsQueueSession_DeleteTemporaryQueue`	`tibemsSession_DeleteTemporaryQueue`

Closed Issues

This section lists issues that were closed in the named releases. Unless otherwise noted, the closed issue affected installations across all platforms.

Reference # and Release	Description
Issues Closed in Release 5.1.5	
1-AF5YQB	A client application with multiple consumers on a single session sometimes did not receive messages from both message listeners. This has been fixed.
1-AO1I78	Fixed an error that sometimes caused the EMS server to increment a message delivery count even though the message was not presented to the consuming application. This problem occurred on routed queues following a session recover or session close call.
1-ANC64L	Fixed an error that caused message arrays exported to SmartSockets to send exactly four elements in the array, regardless of actual array size.
1-AN8LIB	When exporting a nested StreamMessage to SmartSockets, the message was created as an int2_array instead of a nested SmartSockets message. This has been fixed.
1-AIHAZB	Fixed an error that prevented a message acknowledgement from being sent after the message consumer closed. This occurred when an application was consuming from a routed queue.
1-A8CVA9	Fixed an error that caused the EMS server to hang for 60 seconds when a .NET client attempted to connect to the server using a TCP protocol on an SSL port. The .NET client now returns immediately following such an event.
1-A79ZS7	Fixed a problem that caused the EMS server to inconsistently apply name length limitations to destination properties. When adding or setting a property using the administration tool or API, a destination property value would sometimes be truncated when written to the configuration files, and so prevent the EMS server from restarting. The full-length value could be manually added to the configuration files without impacting EMS server startup. The problem has been corrected so that the behavior of the API, administration tool, and configuration files are the same. The length limitation for all destination property values is 64 characters.
1-9YXQRS	Fixed a problem that sometimes caused a warning to appear when the EMS Schema Export Tool was run against a MySQL database on UNIX platforms.

Reference # and Release	Description
1-9K712H	A known issue in JDK 1.6 sometimes caused the 64-bit EMS server to stall and not process persistent messages from clients. This has been corrected. For more information on the known issue in the JDK, see http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6791815.
Issues Closed in Release 5.1.4	
1-9YUW19	The Java client only established an SSL connection if it was able to verify a full chain of trust from the server's certificate to a self-signed root CA certificate. This has been fixed when the client application sets the system property `com.tibco.security.NoExplicitCAChain` to `true`.
1-85L3NL	The Java client would accept a self-signed certificate for the server identity even though the certificate did not match the client's trusted certificates. This has been fixed on all platforms including aix52/power.
1-ACPT3R	Some Java clients were unable to read P12 Entrust certificates, leading to the error: `javax.jms.JMSSecurityException: Error occured while reading identity data: No keys found in keystore`. This has been fixed.
1-9N9D1T	Fixed an error on UNIX platforms that prevented the EMS server from authenticating a user after an LDAP server restart.
1-AB5WS5	The JMSXDeliveryCount was incorrect for XA transactions that were rolled back. This has been fixed.
1-AB5HEJ	Fixed an error that could cause high CPU usage and read rate when using a queue browser and consuming messages from the queue.
1-AA1X8T	Fixed an error that sometimes caused a server error when a client sent a zero-length array inside a MapMessage.
1-A419G0	Fixed an error that sometimes caused an EMS client application to fail with an unexpected `java.lang.ArrayIndexOutOfBoundsException: 2 >= 2` error message when interrupted during connect attempts. This has been corrected to throw a JMSException.
1-A35PT9	Fixed an error that could cause an EMS server to disconnect from clients or other EMS servers. This happened when the EMS server was busy and did not service incoming heartbeats within the `server_timeout_client_connection` (or `server_timeout_server_connection`) period.

Reference # and Release	Description
1-A27UU7	Fixed an error that sometimes caused the EMS server to crash when frequent client disconnects were occurring.
1-9YRCK9	Messages published to a topic and discarded when conditions set by the overflowPolicy were met would sometimes reappear after a server restart. This has been fixed.
1-9TSSHZ	When an external LDAP user was connected to the EMS server, adding the user to a static group did not write a permanent change to the `groups.conf` file. This has been fixed.
1-9RFYC4	Previously, users were only able to supply passwords to the EMS administration tool on the command line. You can now supply passwords using a file.
1-9R9VEL	Previously, routing protocols rendered large hub-and-spoke configurations unstable in some circumstances. These routing protocols have now been optimized, improving the stability.
1-9PCRZE	Fixed a defect that would cause a timestamp update of files `acl.conf` and/or `groups.conf` when the last connection using an external user was closed, and an admin command was issued. The presence of at least one ACL was required for this issue to be observed.
1-9OZ7IN 1-9RY6EB 1-83293T	Fixed an error that under certain circumstances led to a resources leak in the client, and which sometimes prevented EMS clients from reconnecting to the EMS server after a disconnect.
1-9N1OSY	Fixed a problem in the.NET admin client API that caused the Exception Thrown count (as seen by perfmon windows utility) to erroneously increase in certain cases.
1-9HERLH	Fixed an error in java and .NET clients that sometimes caused asynchronous message consumers to receive more than one message after `MessageConsumer.Close()` was called.
1-8U6ROW	Fixed an error that could sometimes cause the EMS server to shut down when a command issued through the administration tool included a selector syntax error.
1-8BTNMN	Fixed an error that caused Java and .NET consumers receiving StreamMessage and BytesMessage to get incorrect data when the message was redelivered to the asynchronous message listener. This happened when the asynchronous message listener previously threw a RuntimeException while processing the message.

Reference # and Release	Description
1-A1HUEA	When using database stores, it was possible for fault tolerant heartbeats to the backup server to be delayed when the primary server was busy processing messages. This has been fixed.
1-A95UIO	Fixed an error that sometimes caused an EMS client to deadlock during a fault tolerant failover while `EXPLICIT_CLIENT_DUPS_OK_ACKNOWLEDGE` mode was enabled.
Issues Closed in Release 5.1.3	
1-A7OAAP	The size of destination objects increased in Software Release 5.0. This has been corrected to decrease the required memory allocation for destination objects.
1-9OZIR8	The EMS server could crash while creating a session when it encounters an error. This has been fixed.
1-A6MJHC	A non-transactional persistent topic publisher sometimes did not receive an exception when there were no subscribers to the topic but the topic was bridged to a failsafe queue and the queue write encountered an IO failure. This has been fixed.
1-9Y1RK7	An internal EMS server XA transaction timeout can occur after the server has prepared an XA transaction, which causes the server to roll the transaction back and violate its contract with the transaction manager. This has been fixed.
1-9Y0M5C	The EMS Server may crash following a route disconnect if messages from that route arrived just prior to the disconnect event. This has been fixed.
1-9XYQ59	Fixed a defect that could cause the server to crash. This sometimes happened when expiring messages were sent to a queue that had pending messages, at least two consumers, and if one of the consumers was closed.
1-9XYBFQ	Fixed an error that could cause a memory leak when a topicSubscriber used a connection with a clientID set, then the connection was then closed and a new connection created.
1-9HAFEF	If the server was shutdown while a queue purge was still in progress, messages would sometimes reappear in the queue when the server restarted. This has been fixed.

Reference # and Release	Description
1-9E9JIN	Resolved a performance issue that impacted the EMS server in situations where a large number of pending messages existed in a queue and parallel producers were sending messages to the queue within XA transactions. Performance has been improved.
1-9YNDON	When using database stores, messages that were swapped out (for example, if no consumers existed or if max_msg_memory limit was reached) and then swapped back in, sometimes caused a memory leak in the server. This has been fixed.
1-9X467X	On a fault tolerant failover, the new active server would sometimes throw the configuration error, "failed to create ACL entry, reason=No memory". The error could occur if the group "all" was referenced in acl.conf. This has been fixed.
1-9ZE4EF	Fixed an error that caused memory growth in the EMS server when using database stores.
Issues Closed in Release 5.1.2	
1-9PZDH0 5.1.2	On a queue with two or more message consumers and where message selectors were used, an unexpected shutdown of one of the consumers would sometimes cause messages in the queue to be left undelivered. The messages would not be delivered unless one of the consumers was restarted. This has been fixed.
1-9PRVH7 5.1.2	Fixed an error that sometimes caused a `java.lang.NullPointerException` error when using database store types with fault tolerance.
1-9PWFZZ 5.1.2	Fixed an error that sometimes caused the backup server to fail and generate a `java.lang.NullPointerException` error. This occurred during a fault tolerant failover when database stores were used and routes were defined in the routes.conf configuration file.
Issues Closed in Release 5.1.1	
1-9OSOOR 5.1.1	Fixed a defect in the EMS 5.1.0 server that sometimes caused a memory leak when processing a connection. The memory leak occurred if the server authenticated the connection request with a password in the configuration files that was created by an EMS version 5.0 or earlier server.
1-9N8R5D 5.1.1	Fixed an error that sometimes caused an EMS 64-bit server, running on Windows 2008, to hang when connecting to more than 60 clients, and throw the error message: "select error: 10038 (The descriptor is not a socket)".

Reference # and Release	Description
Issues Closed in Release 5.1.0	
1-6Y10WH 5.1	If a temporary destination was created, and a destination of the same type with wildcards in the name that would cause it to match the temporary destination was either created or updated, the properties of this "parent" destination would be incorrectly transferred to the temporary destination. This has been fixed.
1-8QUAGB 5.1	When a queue consumer with a selector was started and there were pending messages in the queue, all messages were reevaluated against the selector. This sometimes led to swapping in messages from disk. If a message doesn't pass the selector evaluation, and therefore is not sent to the consumer, it is now swapped back to disk immediately, reducing the overall message memory usage. This also fixes a performance-related issue, where a queue with consumers that had selectors running at different speeds (or a mix of consumers with and without selectors), could reach a level of message memory usage that caused the server to swap messages back to disk.
1-9GWK45 5.1	Fixed an issue that would cause the server to preallocate a store file to a size bigger than the `file_minimum` value. The actual size would be rounded up to a size divisible by 4MB (for instance, `file_minimum` set to 11MB would produce a file of 12MB). Also, the preallocation would not occur if the file size prior to the preallocation would be, once rounded up to a size divisible by 4MB, greater than the `file_minimum` value (for example, if the original file size was 5MB and `file_minimum` was then set to 7MB). Now, the file size after preallocation is the `file_minimum` rounded up to the next 512 bytes, which is the minimum record size used by the server.
1-9HF0WQ 5.1	Fixed a defect that could cause the server to core dump before an exit, if the server failed to open/lock any of its stores, and a connection was made at the same time.
1-9ACGVN 5.1	If destination tracing was enabled after a message had been received from that destination but before the message expired, the trace and/or monitor event sometimes did not include the expired message ID. This has been fixed.
1-9AAJEL 5.1	A fault tolerant application subscribing to a destination's monitoring topic (such as `$sys.monitor.Q.r.`*destination_name*) would stop receiving monitor events after a server failover. This has been fixed.

Reference # and Release	Description
1-995VY3 5.1	Fixed a defect that was preventing changes made to the parameter `msg_swapping` using the administration tool or API from being persisted if the parameter was missing from the configuration file.
1-995XCF 5.1	The predefined group called `all`, which is described in the *TIBCO Enterprise Message Service User's Guide*, was not functioning as documented. This has been fixed. You can grant destination and administrative permissions to a predefined group called `all`, which contains all users. The group is predefined and not persisted in the `groups.conf` file. It can however be referenced in `acl.conf` in order to grant destination permissions to all users. This group does not need maintenance. That is, the administrator does not have to add a newly created user as a member of the `all` group.
1-992QZI 5.1	In the LdapLookupContext in .NET, SSL certificates were not always correctly reconstructed during a lookup, and multiple `ssl_trusted` file names were mishandled. This could happen when XML representations of SSLConnectionFactory objects were stored in LDAP, and SSL certificates were stored in byte representations. This has been fixed.
1-98U7LB 5.1	In a fault tolerant scenario, a client program sometimes stalled while waiting for a message send to complete. This occurred when with fault tolerant connection for sending and receiving detected a JMS exception while parsing an incoming message (for example, if the message was corrupted) and disconnected the connection. If a sender thread then entered a waiting state, expecting the connection to be reestablished, it can result in a deadlock. The waiting sender will block a session close. This problem has been fixed. The behavior of the C client, which previously ignored a corrupt incoming message, now mirrors the Java and .NET client reaction. All clients report the error and disconnect the connection.
1-984HUF 5.1	Fixed an error in the .NET API that caused an "Invalid Metric" exception to be thrown when performing a lookup of a connection factory object (without a load balance metric set) in the LDAP server.

Reference # and Release	Description
1-97Y2FI 5.1	In a fault tolerant scenario where both primary and backup servers were configured to use database stores, and where there was a network partition between the servers, a failover sometimes resulted in both servers attempting to attain a lock on the database files, and both servers exiting. This has been fixed. After a failover, only the server which does not have the lock exits.
1-97YNRI 5.1	Some functions of the EMS C and C Admin API caused memory leaks. The affected functions are: • `tibemsXAResource_GetRMID` • `tibemsXAResource_isSameRM` • `tibemsAdmin_SetCommandTimeout`, when the timeout was less than 5000 • `tibemsQueueInfo_GetDeliveredMessageCount`, when the destination specified was a topic rather than a queue • `tibemsTopicInfo_GetDurableCount`, when the destination specified was a queue rather than a topic • `tibemsTopicInfo_GetActiveDurableCount`, when the destination specified was a queue instead of a topic All functions have been fixed.
1-97SVOR 5.1	Messages with the boolean property `JMS_TIBCO_PRESERVE_UNDELIVERED` set to `false` were sent to the `$sys.undelivered` queue when the message was either expired, or the queue was reaching its `maxRedelivery` value. The EMS server then incorrectly recovered and moved such messages to the undelivered queue on start-up. These two issues have been fixed. Note both the EMS server and EMS clients must be upgraded to EMS 5.1 in order for this fix to be applied. Upgrading just the client or just the server does not fully correct the problem.
1-97JS36 5.1	Fixed an issued that was preventing the EMS Server from sending very large messages to consumers. No specific message size was causing the issue, as it depended on the system resources available at the time of the send. The behavior was primarily observed in EMS servers running on Windows platforms.
1-913TWN 5.1	An issue was preventing an application using the TIBCO EMS C Client from sending and/or receiving very large messages. The message size limit depended on the system resources available, and the error primarily occurred on Windows platforms. This issue has been fixed.

Reference # and Release	Description
1-96Z8RD 5.1	A write error sometimes occurred when attempting to receive messages using a .NET SSL client. This was caused by asynchronous NetworkStream writes in .NET, which in turn caused multiple threads to write to the NetworkStream at the same time. This has been fixed to lock the underlying stream object during a write.
1-964KHF 5.1	The `[Topic]Session.CreateDurableSubscriber` calls of the Java and .NET API, and the `tibems[Topic]Session_CreateDurableSubscriber` calls of the C Client were not checking for a null or empty string for the durable name. This was causing the creation of a durable without a name, which was then impossible to remove using the administration tool. The unnamed durable could only be removed by the unsubscribe() method. This has been fixed in all three APIs. A fix is also present in the server so that it now rejects durables with empty names (from older clients) and discards, on startup, the ones that may have persisted from servers of earlier releases.
1-95Y3CH 5.1	Fixed an error that could cause Java clients to receive a `ClassNotFoundException` when using Object messages. Deserializing an Object message when the object is a class instance that represents primitive types sometimes resulted in an exception.
1-95XJDX 5.1	Setting the parameters `server_heartbeat_client` and/or `server_heartbeat_server` to the value `5` caused a small memory leak when clients and routed or fault tolerant servers were disconnecting. This defect has now been fixed.
1-95JEKP 5.1	If there is no pre-existing bridge for a given source destination and an attempt to create a bridge with a invalid selector is made, the creation of that bridge is correctly rejected. However, a subsequent `show bridges` command would cause the administration tool to crash. This has been fixed.
1-959HFL 5.1	Fixed a defect that could cause EMS applications to report "Invalid magic in the message" when both `ssl_auth_only` and `server_heartbeat_client` were defined on the server.

Reference # and Release	Description
1-94X47E 5.1	When using database stores with routes, there was chance of receiving duplicate messages from the route. This happened if, while the EMS server was waiting for acknowledgements from the route, the message was administratively deleted on the EMS server while the route was disconnected, and the EMS server restarted during the route disconnect. This has been fixed.
1-93NXTO 5.1	Previously, the database store feature experienced some problems when used with MySQL and storing large messages. This problem has been fixed. If you are using EMS database store feature with MySQL, run the EMS Schema Export Tool with an `-updateall` flag in order to change the `large_message_body` column from blob to longblob.
1-9G6CPD 5.1	The database store feature previously limited the size of `selctor_strings` and `selector_bytes` columns to 255 bytes. This limit has been increased to 4000 bytes. If your application has long `selctor_strings` and `selector_bytes` values, update the database tables by running the EMS Schema Export Tool with the `-updateall` option.
1-934FLK 5.1	An error prevented the store file parameter `file_minimum` from being set to `0`. This has been fixed. Setting `file_minimum=0` specifies no minimum disk space preallocation for the store file.
1-92OLHZ 5.1	It was previously possible for a consumer on a non-temporary destination to receive messages sent to temporary destinations. JMS Specifications defines temporary destination as unique destinations. It is therefore incorrect to allow a consumer on, for example, `topic>`, to receive messages that were sent to a specific temporary topic. This has been fixed.
1-92CUL7 5.1	Fixed an error that incorrectly allowed `maxRedelivery=1` to be set in the `tibemsd.conf`. The valid range for the `maxRedelivery` destination property is 2 to 255.
1-927B5O 5.1	A race condition between the deletion of a temporary topic on server A and incoming messages for that temporary topic coming from a routed server B could cause server A to core dump. This has been fixed.

Reference # and Release	Description
1-926SK0 5.1	If messages were consumed using an XA transaction that was prepared and then rolled back, an EMS Server restart could in some cases cause the loss of those messages. This has been fixed.
1-926SIY 5.1	The "Message Size" shown by the administration tool commands `show db` and `show store` was sometimes wrong. For example, the message size could sometimes be a negative value, or a non-zero value when no pending messages existed. This could be observed, for instance, following a server restart and after consuming all pending persistent messages, or purging non persistent messages that had been swapped-out on disk. This has been fixed.
1-926SGS 5.1	In the administration tool, the command `show store` *store-name* reported the used space as free space and vice versa. This has been fixed.
1-923RY3 5.1	EMS 5.0 did not support `HIGH` or `MEDIUM` cipher strings defined in `ssl_server_ciphers`. This has been fixed; `HIGH` and `MEDIUM` strings are now supported.
1-8ZDUQJ 5.1	Previously, the EMS server with routing disabled did not remove all information after the last route in a zone was deleted. This could cause an error if a new route was created in the same zone with a different zone type. This has been fixed.
1-8Z8LLZ 5.1	Fixed an error that sometimes caused C client applications using SSL connections to have some problems disconnecting from the EMS server.
1-8ZDO9H 5.1	When message swapping was disabled, messages that were read from a database store during a recovery included only message headers. Message bodies were not read. This has been fixed.
1-8Z61ZB 5.1	Fixed an error where `tibemsConnectionMetaData_GetProviderVersion` returned incorrect information on some platforms.
1-8YXO0K 5.1	When creating a durable subscription, if a badly formed selector was assigned, the subscription failed but no error was returned. This has been fixed.
1-8YPD98 5.1	Fixed an error in the administration tool that caused the command `show stat route topic=`*topic_name* to give an "Invalid parameter" error.

Reference # and Release	Description
1-8YBI09 5.1	Changed C++ type comments to C style comments in the `emsadmin.h` file.
1-8Y8W7X 5.1	Fixed an error that prevented the `tibemsadmin` tool from rolling back or committing a transaction when the transaction contained unprintable characters.
1-8Y8S0J 5.1	An application using the .NET client API would sometimes hang during the `Connection.CreateSession()` call, if the connection to the server was lost while in the middle of the call. This has been fixed.
1-8XV8AR 5.1	Deleting a queue in the home server sometimes caused a consumer connected to a proxy server and consuming on that queue to stall. A subsequent thread dump would show the consumer waiting for a message acknowledgment. This has been fixed.
1-8X91MY 5.1	In some extremely rare situations, the server would send a corrupted message to the client or routed server. The client would then report a "javax.jms.JMSException: Corrupted incoming data message". The routed server would report a "SEVERE ERROR: No memory to extract routing info from incoming message". This has been fixed. Note that some external corruptions (of the network or disk), unrelated to the issue described here, could still produce the same symptoms.
1-8X2VQB 5.1	If a static durable subscriber was created with a client ID containing spaces, after a server restart, another dynamic durable would be created with a client ID equal to the portion of the client ID that preceded the first space. For example, given a client ID of `client id1`, after a restart the EMS server would create a durable subscriber with the client id of `client`. This has been fixed.
1-8UNBE9 5.1	When `ft_url` is set, the `store_minimum_sync` was not being honored during server startup. This has been fixed.
1-8SDXCE 5.1	Previously, the log message "Access denied for queue browsing" was included in the `PRODCONS` trace option, and printed only when that option was on. This log message has been moved, and is now under the `ACL` trace option.

Reference # and Release	Description
1-7DAUXQ 5.1	Fixed an error that allowed the creation of destination names containing spaces (not enclosed in double quotation marks) using the administration API. When such a destination existed, and was used as the target destination for a bridge, the bridge could not be persisted in the configuration files, which in turned prevented the EMS server from restarting after a failure.
1-6PKJ4A 5.1	Fixed a problem that caused the server to disregard the `JMS_TIBCO_DISABLE_SENDER` message property setting; setting the value to `false` had the same effect as a `true` setting.
1-6MEDYR 5.1	The monitoring topic `$sys.monitor.server.warning` did not receive messages when related events took place. This has been fixed.
1-9B19SN 5.1	Added CHANNELS to the detailed statistics definition in the server.
1-9E5NMN 5.1	Previously, a fault-tolerant .NET message producer could return an exception during failover and not correctly reconnect to the other EMS server in the fault-tolerant pair. This behavior has been fixed.
1-9HERKU 5.1	Fixed an error that affected .NET clients only. Calling `MessageConsumer.Close()` for asynchronous MessageConsumers sometimes caused `MessageConsumer.Close()` to stall and not return, even after the `onMessage` callback returned.
1-91NN3D 5.1	Fixed a problem that adversely affected performance of the server when processing non-persistent messages sent by clients using a transacted session.
1-9IUY4L 5.1	Fixed a problem that sometimes resulted in "server not connected" errors while using Java 1.6, when in fact the connection was fine.
1-9C93O5 5.1	Fixed an error that sometimes caused the EMS server to terminate SSL connections if it encountered a certificate error during verification of a new SSL connection.
1-96PZZI 5.1	Fixed an error that could cause a server to crash when a connection was administratively deleted during a transacted send.

Reference # and Release	Description
1-96YSFI 5.1	Fixed an error where the EMS server could fail when more than 1024 file descriptors were open while it attempted to connect to a peer EMS server using an active route.
Issues Closed in Release 5.0	
1-85L3NL 5.0	The Java client would accept a self-signed certificate for the server identity even though the certificate did not match the client's trusted certificates. This has been fixed on all platforms except aix52/power.
1-8X8G0U 5.0	Fixed an error that prevented temporary topics from being deleted. The `tibrvjms.jar` could cause the EMS server to leak temporary topics in two ways: • If the application used the `sendRequest` method, it created a temporary topic for the reply which was not later deleted. • If the application called the `createInbox` method, the corresponding temporary topic was immediately leaked. This has been fixed. See the documentation for `TibrvJMSTransport` in the *TIBCO Enterprise Message Service Java API Reference* for more details.
1-8W9HMX 5.0	Fixed an issue that could cause a producer to receive the exception "Invalid temporary destination" when sending to a server that had recovered from a failure. This error appeared when publishing to a temporary destination for which no consumer existed. In some cases, messages were successfully sent to a temporary queue, but both queue and messages were removed when the producer's connection closed. Both these issues have been fixed.
1-8NO35X 5.0	Fixed an error that prevented a message from being written to the logfile when log rotation failed. A message indicating the error and the fact that automatic log rotation is disabled was still logged to the console. If the previous log rotation failed to open a new logfile, this error could also cause the server to crash.
1-8JH96Q 5.0	Offline durable subscribers' statistics were not reset as part of the statistic reset procedure. This has been fixed.
1-8HC6OV 5.0	Fixed an error that allowed the EMS server to delete a destination that also served as a bridge. The corrected behavior requires that the bridge be deleted first, before the destination can be deleted.

Reference # and Release	Description
1-8H5D43 5.0	When a Java EMS application used an `XATopicConnectionFactory` or `XAQueueConnectionFactory` and to create, respectively, an `XATopicConnection` or `XAQueueConnection`, the connection appeared as generic in the administration tool. The connection now appears with the same type as the connection factory used to create it.
1-89N7BN 5.0	Fixed an error that caused the server to crash when attempting to bind an incorrect LDAP URL.
1-83I26S 5.0	Fixed an error that caused message delivery to stop when a ServerSession was closed.
1-831L4N 5.0	Fixed an error that prevented EMS clients from connecting to a connection factory that was defined with multiple URLs, and for which the server hostname was not specified. For example, where `url=tcp://7222,tcp://7222`. An EMS client looking-up this connection factory would fail to connect.
1-7DUS3R 5.0	Fixed an issued that caused the bridging of a dynamic destination to stop when the destination was promoted. This failure in the bridge occurred when the dynamic destination was promoted in the administration tool using the `create` command, and when the bridge did not have a parent destination.
1-56UNLN 5.0	Corrected a misleading error message for the administration tool command `delete durable`. The misleading error appeared when multiple arguments were supplied with the command.
1-4ZLF6I 5.0	The C function `tibemsConnection_GetMetaData` was not implemented even though it was defined in the header file. This has been corrected.
1-8VQL2A 5.0	Corrected a problem which caused SSL initialization to fail on AIX. In previous release of EMS, the `tibjms.jar` and `tibjmsadmin.jar` initialized the SSL vendor to `j2se`, which did not work on AIX. In EMS 5.0, these JAR files initialize to `j2se-default`, which selects the IBM SSL vendor when running on AIX.
1-8T2VYQ 5.0	Fixed an error that prevented synchronization between the EMS client and server when the client time was ahead of the server time.

Reference # and Release	Description
1-8SY3TQ 5.0	Corrected an inconsistency in the way the EMS server reacted to errors encountered during recovery. The server's default error recovery policy is now to stop immediately when a corrupt record error is detected. This change is behavior is further described in Changes in Functionality on page 20, and in Error Recovery Policy in the *TIBCO Enterprise Message Service User's Guide.*
1-8NCZB0 5.0	The EMS Server would incorrectly clean up dynamic destinations with messages sent to closed consumers. This sometimes caused exceptions in the client when trying to acknowledge those messages, or message loss when the client application tried to rollback the messages. This has been fixed.
1-8LNJE1 5.0	Fixed an error that allowed the EMS server to restart after the `routes.conf` file was directly edited to make changes to the name and type of a route zone. Changing the zone name or type by editing `routes.conf` could lead to the creation "ghost" consumers for the route, or consumers that did not appear when using the show durables command. The ghost consumers caused the server to store pending messages on the destination for which such ghost consumer existed. On detecting the zone name or type changes, the server now aborts the startup. You must revert the configuration changes and apply the changes through the administration tool or API.
1-8LGBPZ 5.0	Fixed an error that caused the EMS server to crash when an incorrect `ldap_static_group_attribute` was specified in the `tibemsd.conf` configuration file.
1-8JKS7Y 5.0	Fixed a problem that caused C programs to hang when a message consumer tried to close the consumer, session, or connection from within the listener callback. To parallel the Java library, the C API now includes the function `tibems_SetAllowCloseInCallback`, which can be used to prevent a deadlock when closing a message consumer, session, or connection from within the listener callback. See the *TIBCO Enterprise Message Service C & COBOL Reference* for more information about `tibems_GetAllowCloseInCallback` and `tibems_SetAllowCloseInCallback`. In addition to the new functions, the EMS server now displays an error message when it detects a possible deadlock.
1-80E5R1 1-5NJYNG 5.0	In the EMS .NET API, `PropertyNames` for the Message class did not appear up in IntelliSense, in Visual Studio. This has been fixed.

Reference # and Release	Description
1-53D0WX 5.0	Fixed a problem with message acknowledgement that could cause an invalid message acknowledgement to be accepted without throwing any errors. This could result in, for example, a fully acknowledged message reappearing in a queue after a recovery. The EMS server now verifies that incoming acknowledgements are valid. This change prevents the mishandling of acknowledgements in the following situations: • The acknowledgment arrives after a message has expired. • Following a fault-tolerant switch, a routed queue consumer attempts to acknowledge a message that has been reassigned to another consumer. • Following a fault tolerant switch, a consumer attempts to acknowledge a message that was received prior to the failure, but which in the meantime was received and acknowledged by another consumer.
1-8RET20 5.0	Fixed an error that could cause the EMS server using LDAP authentication to crash when a client attempting to connect loses its connection before the server processes the request. This occurred only when the `user_auth` parameter contained the value `ldap`.
1-8PG1IA 5.0	Fixed an error that could cause the EMS server to hang when the administration tool attempts to shut it down.
1-8L9RV5 5.0	Fixed defect that would crash the server if an administrative command, such as `create jndiname`, attempted to create very long lines in the configuration file. The server now returns an error when the line is too long for the configuration file.
1-7UMDSB 5.0	The EMS server did not report a startup error when a `client_heartbeat_server` interval was not set with the `server_timeout_client_connection` limit. This has been fixed. If you do not set the `client_heartbeat_server` parameter when a `server_timeout_client_connection` is specified, a configuration error is generated during startup. If `CONFIG_ERRORS` is part of the `startup_abort_list`, the server will not start. If not, the error is printed but the server starts, and clients will be disconnected after `server_timeout_client_connection` seconds.
1-8HCBEJ 5.0	Fixed an error that prevented a server in a chain of servers from receiving messages from other servers when it subscribed using a global wildcard. For example, in a chain of servers A, B, and C, a wildcard subscriber to `foo.>` on server C, would not receive messages sent to a matching topic on server A.

Reference # and Release	Description
1-8D1Q54 5.0	Fixed an error that prevented a clientID from restarting after a failover when the connection was created using `Connection.setClientID()`.
1-8CRA2H 5.0	Fixed an error that caused an unlimited prefetch size to be set for a topic, global, or routed queue if `prefetch=none` was specified. Because `prefetch=none` is not supported for these destinations, this setting now prints an error, and prefetch is set to the default size.
1-8CJM4K 5.0	Fixed an error that allowed EMS client applications to perform JNDI lookups on dynamic destinations. Now, if a client attempts to look up a dynamic destination, the status Not Found is returned.
1-8BSRY3 5.0	Corrected an inconsistency between the message timestamp that was shown in the server log (when destination tracing was enabled) and the message timestamp printed by the client. The two timestamps are now equal.
1-8B90XP 5.0	Fixed a problem that caused C applications to crash when `tibemsConnectionFactory_CreateConnection` was called on a Connection Factory that had specified SSL but did not specify any SSL parameters. Note that specifying SSL without any parameters is invalid and so this function will still fail but will not crash.
1-8B1TSG 5.0	Corrected the behavior of the C function `tibemsDestination_Copy`, which incorrectly allowed temporary destinations to be copied. Temporary destinations are unique and cannot be copied.
1-87R9M6 5.0	Fixed an error that caused `ConsumerInfo.Details.getTotalAcknowledgedCount` and `ConsumerInfo.Details.getTotalMsgCountSentByServer` to return incorrect data.
1-84MRFP 5.0	Fixed a defect in the C API that could cause `tibemsErrorContext_GetLastErrorString` to return random characters if the connection failed to complete before the connect attempt timeout.
1-84G941 5.0	The Java client did not use the `com.tibco.tibjms.connect.attempttimeout` to set the connection attempt timeout value for `FACTORY_CONNECT_ATTEMPT_TIMEOUT`. Now it does.
1-848ZER 5.0	Fixed an error that prevented the system property values for connect attempt, delay, and timeout from taking precedence over a connection factory's settings.

Reference # and Release	Description
1-842XI6 5.0	Corrected the error that is thrown when two groups with the same name are added to the `groups.conf` file. The server will now reject the creation of the second group.
1-83Z45V 5.0	Corrected a problem that caused the administration tool to display incorrect error messages with the `showacl` command.
1-83LY19 5.0	Fixed a problem that prevented the use of the `setprop queue` command to set properties for a routed queue.
1-823HWG 5.0	Fixed an error where a C client that requested SSL only for authentication would receive encrypted data after a failover of the EMS server.
1-8B6GW3 5.0	The `tibemsMsgConsumer_Receive` function in C now correctly returns `TIBEMS_INTR` when it is interrupted because the EMS server is shut down. In EMS 4.4, the error returned from this condition had changed to `TIBEMS_EXCEPTION`, creating a minor backward compatibility issue.
1-8AHWB3 5.0	Previously, there was no function available to get a certificate store from the connection factory object. A new .NET API, `GetCertificateStore`, was added to the connection factory.
1-83ZV35 5.0	The .NET SSL connection failed when the client certificate was stored in the Microsoft Certificate Store. This was caused by the way the certificate lookup was handled; a new API, `SetCertificateNameAsFullSubjectDN`, has been added to `EMSSSLSystemStoreInfo` in order to correct the problem. `SetCertificateNameAsFullSubjectDN` should be used in place of `SetCertificateName`.
1-80YOHX 5.0	The `event_reason` field was missing from `$sys.monitor.connection.connect` monitor messages. This has been fixed.
1-80OO9I 5.0	Fixed a defect that could cause a memory leak when encoded messages were swapped in and out of memory.
1-7TISJA 5.0	Fixed an error that could cause duplicate monitor and trace events to be reported while the EMS server is recovering store files after a restart.

Reference # and Release	Description
1-6X34EJ 5.0	Fixed an error that caused the command `show stat producers` to return producers that had not sent any messages to a named destination. This could occur when the `detailed_statistics` parameter contained `PRODUCERS`, and at least one unidentified producer, or producer created on the destination `NULL`, was connected. Entering `show stat producers queue=`*queueName*, for example, would return a list of producers that included the unidentified producers, even if they had not sent messages to *queueName*.
1-6SI03T 5.0	When using the Java or .NET administration API to purge a destination, a user without purge permissions did not receive an error indicating that the destination was not purged. An attempt to purge a wildcarded destination also did not generate errors, although the `purgeQueue` and `purgeTopic` methods do not permit wildcards. This has been fixed; errors are now generated in these instances. In EMS 5.0, the Java and .NET libraries include new methods that provide "purge all" and "delete all" functionality: `purgeQueues`, `purgeTopics`, `destroyQueues`, and `destroyTopics`. See the HTML documentation for Java or .NET libraries for more information.
1-80V8A3 5.0	Fixed an error that could cause a Java client to hang when creating a browser, if an administrator enabled authorization after the client created the connection to the EMS server, but before `createBrowser()` was called. This occurred only when the connection creation would have failed if authorization had been enabled when the connection was created.
1-80UKUD 5.0	The C library `libtibemslookup` had a dependency on the private LDAP function `ldap_pvt_tls_set_option`, which is not supported by all versions of LDAP. This dependency has been removed. The library now depends only on the public `ldap_set_option`.
1-7M2BWK 5.0	Added the API functions `tibemsMsgConsumer_GetDestination` and `tibemsMsgProducer_GetDestination`, which were missing from the C client.
1-6YHA0P 5.0	Fixed an error that prevented the EMS server startup from aborting when the `startup_abort_list` included `CONFIG_ERRORS`, if the misconfigured parameter was defined before the `startup_abort_list` parameter. Server startup will now fail regardless of the location of the misconfigured parameter.

Reference # and Release	Description
1-8081U9 5.0	Fixed an error that prevented the `JMSXDeliveryCount` property from incrementing as expected: • After a server restart, `JMSXDeliveryCount` did not increase to reflect new redeliveries. Upon a restart of the EMS server, all recovered messages are flagged as redelivered, and their `JMSXDeliveryCount` will be equal to 2, regardless of the `JMSXDeliveryCount` setting prior to the restart. The error has been fixed so that the delivery count is correctly incremented from 2 following the server restart. • The server incorrectly incremented the `JMSXDeliveryCount` for messages received by a consumer and then resent. This behavior has changed so that the resent message is not flagged as redelivered, and so does not include the `JMSXDeliveryCount` property.
Issues Closed in Release 4.4.3	
1-8QX8WM 4.4.3	Fixed a problem that caused C clients using `tibemsMsgConsumer_ReceiveTimeout` to encounter some latency when the receive call times out at the same time a message is received.
1-8WOLDR 4.4.3	Fixed an error that prevented offline durable subscribers from correctly evaluating message selectors.
1-8SNDCE 4.4.3	Fixed a defect that caused QueueBrowser to stop browsing messages when a message had an expiration set, or return messages that should have expired. This problem occurred when using the Java, C and .NET API in EMS 4.4.1 or 4.4.2.
1-8SYUZ1 4.4.3	Fixed a defect that prevented reconnection to the server following a disconnect from a connection with a clientID that encountered out of order XA actions, such as XA rollback before XA end.
1-8SZ0ZF 4.4.3	Fixed a defect that caused a C client to crash during a session create call if the link reader detected a server disconnect and attempted to clean up the link.
1-8VTH83 4.4.3	Fixed a defect that caused the EMS server to hang when under heavy stress conditions involving the import of RV messages.
Issues Closed in Release 4.4.2	
1-8PD8D9 4.4.2	Fixed a defect that could cause the C client to report that it received an invalid message when it reconnected to a server in auth-only mode.

Reference # and Release	Description
1-8P8UMF 4.4.2	Fixed a defect that could cause clients to mishandle a message on a queue with `prefetch=none`, when receiving with a timeout. This could result in a message delivered to the client library but not presented to the application.
1-8CGNRG 4.4.2	Fixed a defect that caused the EMS server to recover invalid messages after a server restart. This occurred when an XA transaction was left in prepared state and rolled back after a server restart. Uncommitted messages were recovered following the server restart.
1-8MHVPE 4.4.2	On recovery, the server sometimes crashed when processing acknowledgement records for messages with the undelivered property and expiration time in the past. This has been fixed.
1-8O2EA9 4.4.2	Fixed a defect that could cause the EMS clients deadlock when acknowledging from a different thread, if the consumer queue had `prefetch=none`.
1-82HYZW 4.4.2	In EMS 4.4.0, messages in the `$sys.undelivered` queue that are consumed are not removed from the "Pending Message" count. This has been fixed.
1-820SBH 4.4.2	Fixed a defect that could cause the EMS Server to crash when it would try to discard `NON_PERSISTENT` or `RELIABLE` messages that have reached their maximum delivery count based on the queue's `maxRedelivery` property. Messages with the `TIBCO_PRESERVE_UNDELIVERED` boolean property would be unaffected and are correctly moved to the `$sys.undelivered` queue.
1-81UJTJ 4.4.2	Under an operational mode that disables batched synchronous writes, the server may return the debug message "DEBUG: Insufficient buffer - SEVERE ERROR: Failed writing message". This has been fixed.
1-838PFR 4.4.2	A change in Release 4.4.0 resulted in a performance issue with the data store for valid messages. This has been fixed.
1-7W1MX1 4.4.2	The error "Error: Global or routed queue cannot specify prefetch=none" was returned when using `addprop` to set any property. This has been fixed so that properties (except for `prefetch=none`) can now be set on global or routed queues.
1-837ZMF 4.4.2	If an application that uses a client ID disconnects or terminates before a `commit()` succeeds, immediately reconnecting was not always successful. This has been fixed.

Reference # and Release	Description
1-81ZZCH 4.4.2	Fixed a defect that could cause the server to not recover valid messages if, prior to the restart, the server ran out of memory at a very specific point in time while creating an internal record that is later persisted on disk. This memory failure may not be reported if the user is not using `reserve_memory`. Of course, users should always ensure that the EMS Server has appropriate resources. You can limit the memory usage by setting the `max_msg_memory` parameter according to the system's memory availability.
1-84ZA3C 4.4.2	Fixed a defect that caused the EMS Server to stall and not resume when a server processing incoming messages reached memory limits. Flow stall when a downstream server reaches memory limits is expected behavior. However, message flow should resume after the error condition is cleared, such as when messages are purged and memory usage drops below the limit.
1-85GCU1 4.4.2	Fixed an error that prevented the removal of a temporary queue, created as a reply-to destination between routing partners, after the sender and receiver connections are destroyed. Queues are now removed at regular 3-minute intervals.
1-873JVN 4.4.2	Messages that were published to a dynamic queue and that were part of an XA transaction and in a XA prepared state when recovered by the EMS server, were not being delivered to queue consumers even after a successful commit of the XA transaction. This has been fixed.
1-885OZV 4.4.2	Messages that were received by a queue consumer as part of an XA transaction, and which were recovered in an XA prepared state during an EMS server restart, did not appear in the dynamic queue. This has been fixed.
1-8B1VDW 4.4.2	In .NET clients, `LookupContext` did not correctly set the connection factory parameters for: • `connect_attempt_count` • `connect_attempt_delay` • `connect_attempt_timeout` • `reconnect_attempt_count` • `reconnect_attempt_delay` • `reconnect_attempt_timeout` This has been fixed.
1-8B1VFQ 4.4.2	Removed unused constants from `LookupContext` in .NET.

Reference # and Release	Description
1-8AE5AJ 4.4.2	In a .NET client, a lookup by `LookupContext` returned an invalid SSL connection factory object when `client_identity` was not set. The client was unable to create a connection with the returned object. This has been fixed.
1-8AHWCD 4.4.2	The .NET client did not manage SSL passwords in the same way as a Java client. The .NET client has been updated to allow SSL passwords to be set either globally or through the `createConnection`(*username*, *password*) API.
1-8ARYMZ 4.4.2	Added the ability to override the connection factory properties in the .NET API.
1-866WO5 4.4.2	Durable subscribers failed when the connection was restarted immediately after a previous connection closed. This has been fixed.
1-85NQ4I 4.4.2	When a durable subscriber was created using the administration tool, with the properties `clientid` and `selector`, the `clientid` property was not stored in the `durables.conf` file. On EMS server restart, duplicate durable subscribers were created. This has been fixed.
1-8CQMCB 4.4.2	The server or C library could crash if TX tracing was enabled when it processed an XID with a large number of non-printable characters. This has been fixed.
1-8CQWNO 4.4.2	The `tibemsadmin` tool could not commit or rollback an XA transaction if the XID had a large number of non-printable characters. This has been fixed.
1-8B0VFG 4.4.2	The EMS server could crash if a flow stall occurred on a routed dynamic topic while the server was removing the topic. This has been fixed.
1-8EEBB3 4.4.2	Fixed a problem where fault-tolerant connection consumers (on queues) failed to receive messages after switching to the backup server.
1-8EYRZV 1-8EYS0R 4.4.2	If the export property of a wildcard topic was changed, and a bridge topic inherited the export property, the server did not notify the other servers. This has been fixed.
1-8ASFJ4 4.4.2	The server sent an unnecessary protocol message when the list of subscribers to a destination bridge topic changed. This has been fixed.

Reference # and Release	Description
1-84JMOT 4.4.2	The server sent an unnecessary protocol message when it received a routed message, if consumers were subscribed to the topic but the message did not match the consumers' selectors. This has been fixed.
1-8IK170 4.4.2	Fault-tolerant Java clients hung after switching to the backup server, if the client received an expired message.
1-8N42I2 4.4.2	A server that experiences a disconnect of a passive route may not clean up the thread associated with that route.
Issues Closed in Release 4.4.1	
1-7XUP5B 4.4.1	Fixed a defect that would cause the EMS Server to ignore JMS Headers as Identifiers in selectors for bridges or routes.
1-7ZVETP 4.4.1	Fixed a defect where tibjmsd crashes when importing SmartSockets GMD messages to a failsafe destination.
1-7ZL0K3 4.4.1	Fixed a defect where, under certain circumstances, closing a .NET consumer while there are still unacknowledged messages produced the exception: "System.InvalidOperationException: Collection was modified; enumeration operation may not execute." This situation occurred when the .NET consumer used explicit acknowledgement and the consumer is either consuming from a routed queue or the consumer is connected to an EMS server that is not "consumer close compliant" (for example, any server prior to EMS Release 4.2).
1-808IPH 4.4.1	Fixed a defect where closing a consumer from another thread while the consumer was receiving a message caused the client to hang.
1-7VWCWH 4.4.1	During the commit() operation, when messages in a transaction exceeded a limit, the following behavior has been corrected: • with local transactions, "storage failure" was incorrectly reported to clients • with XA one phase transactions, the server could crash
1-80BSLQ 4.4.1	Stopping and restarting a client before it has reached the XAend call resulted in a "Clientid already exists" error in Release 4.4.0. This behavior did not occur in previous releases of TIBCO Enterprise Message Service. This has been corrected in Release 4.4.1.

Reference # and Release	Description
1-7Z6EBN 4.4.1	Fixed a defect that, under certain conditions, caused the server to fail to start when attempting to recover if the server was purging records during a crash.
1-7Z37GJ 4.4.1	The `Session.close()` method now no longer throws an exception ("Invalid Session") when more than one thread calls the method on the same session. The behavior is now consistent with the JMS specification that requires that the `Session.close()` method can be called concurrently, and that a closed session should not return any exception.
1-7XR9XF 4.4.1	Fixed a defect where TIBCO SmartSockets message type export fails for messages with a negative message type number.
1-7VWCXN 4.4.1	Fixed a defect where, under certain conditions, messages processed at the same time as a failed transaction were also incorrectly failing.
1-7XQZNS 4.4.1	Fixed a problem in which, if a factory contains a certificate file name that cannot be resolved by the EMS Server, a C client looking-up this connection factory would fail with status `TIBEMS_NO_MEMORY`. The status returned is now `TIBEMS_FILE_NOT_FOUND` and the `tibemsErrorContext` string is "File '<file name>' not found or access denied".
1-7XP1II 4.4.1	Fixed a problem in which the server would recover uncommitted XA transactions after the client had been stopped.
1-7XOSD1 4.4.1	Fixed a defect that could cause the C client that looks-up a Connection Factory containing a certificate file name (`ssl_identity`, `ssl_trusted`, etc..) to return `TIBEMS_INVALID_FILE`. The C Client was incorrectly trying to locate the file locally instead of using the file data sent by the EMS Server.
1-7XM0JP 4.4.1	Fixed a defect in which the EMS server would crash when doing an `xacommit` and the transaction size exceeded the `max_disk_batch_size` limit.
1-7WZ01R 4.4.1	Fixed a defect that, when two servers were configured for fault-tolerant operation, would either allow the active EMS Server to accept a connection from a standby server with a different server name, or to accept a new connection from a second standby server while the first standby server was already connected. Upon disconnection of the second standby server, and after the `ft_activation` interval, the first legitimate standby server would log that it is missing heartbeats from the primary and try to become active. This would repeat until the connection to this standby server is administratively deleted from the active EMS Server.

Reference # and Release	Description
1-7WMAXO 4.4.1	Fixed a race condition that could cause an EMS Server running out of file descriptors to crash.
1-7VOZHR 4.4.1	Fixed a defect that could cause a memory leak in the EMS Server when browsing the last message of a queue while all messages from that queue have been delivered to consumers but not yet acknowledged.
1-7VH178 4.4.1	Fixed a defect in which the `MaxBytes` property was not enforced for non-durable topics when the subscriber was slow in consuming messages.
1-7UR5WP 4.4.1	Fixed a defect that would cause LDAP authentication failure with an error message such as: "ERROR: Missing ldap credential parameter" if the `ldap_credential` parameter contained a password in clear text with less than 5 characters.
1-7Y8VZI 4.4.1	Fixed a defect in the C API that caused the selector passed to the `tibemsQueueSession_CreateBrowser()` function to be ignored, resulting in the queue browser getting all messages present in the queue instead of the ones matching the selector.
1-7Y466T 4.4.1	Fixed a defect in the .NET API in which asynchronous consumers using the `EMSMessageHandler` delegate received only the first message sent when `prefetch=none`.
1-7Y2WFY 4.4.1	Fixed a defect on OpenVMS platforms that, when compiling, generated warning messages for two functions, reporting no "(void)" within the headers.
1-7Y3ACC 4.4.1	Fixed a defect in the C API that could cause a crash if the application attempts to delete a temporary destination that has been already deleted or is deleted by the `ExceptionListener onException` callback.
1-7VERJN 4.4.1	Fixed a defect in the C API that, when SSL is used, receiving a message bigger than 16KB returned `status=5 TIBEMS_EXCEPTION`.
1-7UJHOE 4.4.1	Fixed a problem in the C API in which a call to `tibemsMsg_SetDestination` did not reset an existing destination to Null.
1-7UJHBH 4.4.1	Fixed a problem in the C API in which a call to `tibemsMsg_SetReplyTo` did not reset an existing `replyTo` destination to Null.

Reference # and Release	Description
1-7U53SB 4.4.1	Fixed an error in the *EMS_HOME*`/clients/c/include/tibems/emserr.h` file, which identified the `__cplusplus` constant with only one underscore.
1-7TIXVD 4.4.1	Fixed a defect in the Java admin API, where setting the `reconnect_attempt_timeout` property on a connection factory failed because it was attempting to set the `connect_attempt_timeout` value.
1-4GLF14 4.4.1	If a C lookup to an LDAP server is failing, the user can now get more information using the `tibemsErrorContext_GetLastErrorString()`, which includes the LDAP error code and text for the failed operation.
1-7YCAVP 4.4.1	Fixed a defect that prevented the admin tools and Java/.NET admin APIs from resetting the user's password to null after it was set.
1-7X15QA 4.4.1	Fixed a problem in the tibrvjms.jar file to remove an instance of the `TibrvListener` class that was incompatible with the `TibrvListener` class in the `tibrvnative.jar` file.
Issues Closed in Release 4.4	
1-782QQA 4.4	Fixed a server defect in which the child queues may not inherit the RV import property of a parent queue created after the child queues.
1-77RCVL 4.4	Implemented a new protocol between routes in how they negotiate and work with global/routed queues for improved performance.
1-79SICH 4.4	Fixed a defect that would cause a `BytesMessage` with no body sent by a C client prior to 4.4 to be received as a `BytesMessage` with a body size of 128 bytes by Java or .NET clients.
1-79KS3T 4.4	The Java client's "storage failure" exception during a send is now associated with a corresponding server trace message.
1-79JRZT 4.4	Fixed a problem where the acknowledged monitor message contained an incorrect application message (in the `message_bytes` field) when the application message was swapped out.
1-78IKW5 4.4	Fixed a defect that could cause the EMS server to keep messages that have been swapped out on disk in the datastore when the consumer is using the NO_ACK acknowledgement mode.

Reference # and Release	Description
1-78AH2F 4.4	Fixed a problem where the C client returned an "Invalid port" status when it attempted to create a connection with a load-balanced URL.
1-785J6S 4.4	Fixed a defect that could cause imported RVCM messages to be delivered out of their CM sequence if issues occurred while importing them as JMS messages.
1-785J6C 4.4	Fixed a defect that could cause the server to abnormally exit if a destination with the RV import property was actively importing messages and at the same time was deleted by the administrator.
1-778WRT 4.4	Fixed a problem where, if the server fails to create a transport, the remaining transports specified in the `transport.conf` file were not created.
1-76UIB9 4.4	Fixed a problem where, if a client creates connections to different servers, the `dups_ok acknowledge` mode may not work correctly.
1-752VAO 4.4	Upgraded the Administration Tool and admin APIs to show heartbeat and connection timeout parameters.
1-7494TN 4.4	Fixed a problem where, if a fault-tolerant client detects connection break during a connection->close, a deadlock can occur in the Java client library.
1-744VX2 4.4	Clarified inheritance rules for `OVERFLOW_REJECT_INCOMING` policy.
1-742FBI 4.4	Upgraded sys monitor messages to carry correlation id and `replyto` fields as properties.
1-734840 4.4	Fixed a defect that would cause a route or fault-tolerant connection to fail during the connection initialization if the remote daemon was delayed for several seconds.
1-72BEAW 1-72BEA7 1-1UHRA8 4.4	Upgraded the clients and server so that temporary destinations are fully JMS compliant.

Reference # and Release	Description
1-72BAVQ 4.4	Fixed a defect that would cause a fault-tolerant consumer client to reject messages sent across a route.
1-70UOZF 4.4	Fixed a defect that would cause a queue consumer using a selector in the presence of mixed-priority messages to receive messages that do not match its selector.
1-70N9YG 4.4	Fixed a defect that would cause a sending server to accumulate messages for a receiving server, even when a non-matching `outgoing_topic` selector is defined that does not match those messages.
1-70A8V5 4.4	Fixed a defect that would cause flow control on queues to break after a fault-tolerant switchover.
1-700IB5 4.4	Fixed a defect where restarting a server after purging `$sys.undelivered` could also remove non-purged messages from `$sys.undelivered`.
1-6ZXNIR 4.4	Fixed a defect that would cause the Administration Tool to ignore 'admin' when specified in a `showacl` command.
1-6ZSOU3 4.4	Fixed a defect in which the Administration Tool did not correctly change the permissions for a group in the `acl.conf` file.
1-6Z6AI1 4.4	Fixed a defect in which topic subscribers using fault-tolerant URLs reconnecting to the same server would stall.
1-6Z097Q 4.4	Fixed a defect whereby creating a queue with the exclusive property would produce "Error: Illegal to specify exclusive property for routed queue" if the queue name contained the character 'Q'.
1-6Y66PT 4.4	Fixed a defect whereby a queue browser could not browse dynamic queues or temporary queues created by the same connection.
1-6X8ISQ 4.4	Introduced a new implementation for routed queue acknowledgments so that, When a client consumer sends an acknowledgment for messages received from a routed queue, the EMS server waits until it gets a response from the routed queue's home server before sending the response to the client.

Reference # and Release	Description
1-6X87WF 4.4	Fixed a defect in the C API that would cause calls to `tibemsMapMsg_Set*Array()` functions to return an inconsistent status code when invalid arguments were provided. Now, if the array value is NULL and/or the count is negative or 0, the status `TIBEMS_INVALID_ARG` is returned.
1-6WQM5J 1-C5RMM 4.4	The client library has been updated to use the EMS server time to determine whether a message has passed its expiration time.
1-6WO3RK 4.4	Fixed a problem that, in a fault-tolerant configuration, if the consumer clients were suspended and the primary server shut down, the backup server did not reconnect all consumers when they were un-suspended.
1-6WJRPH 4.4	Fixed a defect where a C client with multiple threads attempting to create a connection simultaneously could receive a 'Server not connected' status in one or more of those threads.
1-6W625F 4.4	A large number of unacknowledged messages used to take a long time to handle when a server fails and the messages are taken over by a standby server in fault-tolerant mode. The performance of this situation has been improved, but the server could still take some time to handle the messages, depending upon how many messages must be redelivered and the speed of the CPU of the failover server.
1-6VASSK 4.4	Fixed a Java defect in which constructor `TibrvJMSTransport` threw a `NullPointerException` when `serverURL` was null and `emulateReconnect` was `true.`
1-6THU18 4.4	The EMS server upgraded to maintain separate counters for normal clients and programmatic admin clients. Normal clients can be limited by setting `max_connections` in the `tibemsd.conf` file.
1-6P48XH 4.4	EMS now supports local transacted sessions for Application Servers using the JMS `ConnectionConsumer` facility.
1-6OZRWR 4.4	`TibjmsXATopicSession` and `TibjmsXAQueueSession` now return `true` for `getTransacted().`
1-6NSMAN 4.4	An `javax.jms.IllegalStateException` exception is now thrown when `Session.createProducer` and `Session.createConsumer` are called with an inappropriate Destination object.

Reference # and Release	Description
1-6NJA1X 4.4	The server has been upgraded to include a transaction suspended state that can be seen by means of the Administration Tool's `show transactions` command or the admin API. Suspended transactions can be rolled back. but cannot be rolled forward (committed).
1-6N8ZG4 4.4	Fixed a problem in which, if a consumer is created on the temp queue and then closed, the temp queue may get deleted while messages are sent to that queue.
1-6G8RT5 4.4	Administration clients were incorrectly reporting that routed queues with no explicit prefetch setting were not inheriting their settings. Also, if another queue's configuration changed then the routed queue's configuration was changed to an explicit prefetch=64. This has been fixed.
1-6B1QV5 4.4	Fixed a problem where the Mac OS X `tibemslookup` libraries had a dependence on a private LDAP function not supported by version 10.4.
1-6AMR79 4.4	The Admin APIs have been updated so that it is now possible to see if a queue is routed and which route it comes from.
1-6A94JD 4.4	`DestinationInfo.getJNDINames` now returns an empty array, which is consistent with Java conventions.
1-5I3CQ3 1-5I3CPN 4.4	Fixed a defect where `meta.db` could grow in size if the server was purging temporary queues.
1-5F00CM 4.4	Fixed a defect that would cause large values for `max_msg_memory` to be incorrectly displayed.
1-5AKLYZ 4.4	A large number of temporary topics (for example, 20000) causes the administration client to crash when performing a `show topics` command. This has been fixed.
1-56UNLG 4.4	The Administration Tool has been upgraded with a carriage return on the end of the error message so that the prompt appears on a new line.

Reference # and Release	Description
1-4VT76R 4.4	Fixed a problem in the C API in which rapidly closing / recreating connections caused an unexpected exit.
1-4UQ5C0 4.4	In C clients, durable subscribers can sometimes cause a deadlock situation when running servers in fault-tolerant mode and failover occurs. This has been fixed.
1-4UPZ2R 1-5N0ES1 4.4	Fixed a problem with the C API if `tibemsMsgProducer_Close` is called after `tibemsSession_Close`.
1-48HEYH 4.4	Upgraded server to restore the `clientID` on recovery. Also fixed a defect where, during fault-tolerant reconnect, the server could reject the connection with a given clientID because it had not yet detected the connection break.
1-3ZOURV 4.4	Server upgraded to check the value of `max_msg_memory` to verify that it is within legal boundaries.
1-3Q8Z7O 4.4	Fixed a defect that could cause a pair of fault-tolerant EMS Servers started at the same time to deadlock while each was trying to enter standby mode.
1-2LP583 1-2ZP5N5 4.4	The server now correctly removes a temporary destination when it is closed and deletes any messages in the destination.
1-226ZVD 1-26CRSU 4.4	Fixed a problem with the queue receiver where `prefetch=none` would stall if the receive method timed out just as a message arrives. Queues with `prefetch=none` now work correctly.
1-1TQTK1 4.4	Closing the session no longer affects the functioning of temporary destinations.
1-6ZXRVP 4.4	Fixed a problem in which quickly exporting large amounts of data to Rendezvous could deadlock the EMS server.
1-7PMMR0 4.4	Fixed a defect where the .NET API did not close the `MemoryStream` object if the `ObjectMessage` failed to deserialize.

Reference # and Release	Description
1-51Q3OY 4.4	Fixed a memory leak while running a multi-threaded C client with 2 durable subscribers.
1-6SUT03 4.4	Fixed a problem that, during heavy usage in a multiple thread environment, caused the C EMS client library to crash or hang upon closing a consumer.
1-77RCM1 4.4	Fixed a defect that would cause the Microsoft Service Control Manager to report in the Event Log that "the EMS Server service hung on starting." This would occur if the EMS Server was registered as a service and configured with the "Automatic" startup mode, and the machine was rebooted.
1-78A3WM 4.4	Fixed a problem in the EMS server that made it possible to send a message to an invalid destination name.
1-78KSCL 4.4	Fixed a problem in the EMS .NET API in which load balanced URLS were not being parsed correctly.
1-79K93T 4.4	Fixed a problem with `JMSTimeStamp` in which C clients, interacting with Java and .NET clients, did not include milliseconds in the timestamp of the received message.
1-7BKC7P 4.4	Fixed a problem in which messages sent to a queue configured with `prefetch=none` and `maxRedelivery=`N were not received N times by a consumer that received in a loop and did an XA Rollback after the XA prepare phase.
1-7C7HP0 4.4	Using start TLS, the EMS server could fail to initialize, but then later connect to an LDAP server after multiple attempts. This has been fixed.
1-7C7L12 4.4	Fixed a defect that would prevent dynamic destinations from inheriting the `expiration=0` property set to the parent destination.
1-7D4F9N 4.4	Fixed a problem in which the client library would hang when using a message listener on a destination with auto-acknowledge when the server is shut down.
1-7DDGLM 4.4	Fixed a problem in the .NET API in which an exception would be thrown when a slash "/" was specified within message selector string.
1-7DPVSB 4.4	Fixed a problem in the C API in which the `tibemsMsgRequestor_Request()` function sent the request message with non-persistent delivery mode.

Reference # and Release	Description
Issues Closed in Release 4.3	
1-5Y3HBI 4.3	Upgraded to zlib 1.2.3 (July 18, 2005), which addresses security issues in zlib 1.2.1. For details about the zlib issues, see these notices: • http://www.kb.cert.org/vuls/id/680620 • http://www.kb.cert.org/vuls/id/238678
1-6NYGF2 4.3	Fixed a server defect in which trace messages concerning routed topics were erroneously categorized as warnings rather than route debug messages.
1-6N8NM5 4.3	Fixed a server defect associated with expiration override and queue browsers. in which the use of a queue browser could erroneously undo the effect of a destination override on individual message expiration values.
1-6N6DQ7 4.3	Fixed a server defect in the Linux 64-bit server in which configuring it to send trace output to *both* a console and a logfile would cause the server to exit immediately at start.
1-6MA3V3 4.3	Fixed a server defect in which the files `meta.db` or `sync-msgs.db` could grow without bound.
1-6J8CDY 4.3	Fixed a server defect in which the server did not properly store the value of the property `ssl_auth_only`.
1-5VHKG1 4.3	Fixed a server defect in which the file `async-msgs.db` would erroneously retain system messages and monitor messages.
1-6MGI0C 4.3	Fixed a server defect in which consumed messages were not properly removed from the server, resulting in memory growth.
1-6NT0OJ 4.3	Fixed a server defect in which disk errors while writing acknowledge messages to store files resulted in messages that were inappropriately redelivered each time consumers would restart.
1-6O3OU3 4.3	Fixed a server memory leak associated with destination monitoring.
1-6O891Y 4.3	Fixed a server defect in which the server erroneously sent monitoring messages to a topic that had no subscribers.

Reference # and Release	Description
1-5JW4Q5 4.3	Fixed a server defect in which topics with transaction semantics did not respect destination limits, such as `maxBytes`.
1-6QTY2D 4.3	Fixed a server defect in which abrupt server exit is associated with expiring messages that remain unacknowledged after their consumer has closed.
1-6QVVB1 4.3	Fixed a server memory leak associated with expiration of messages after a consumer has closed.
1-6Q5ZHU 4.3	Fixed a server defect related to abandoned SSL connections; the server now removes inactive connections after a timeout period expires.
1-6QCFM1 4.3	Fixed a server defect in which removing a large number of messages could trigger a race condition that slowed other server functions and other processes on the server host.
1-6QTIUK 4.3	Fixed a server defect in which a destination property value with incorrect type resulted in memory errors and abrupt exit. For example, null as the value of a string property (such as `sender_name`) could trigger this symptom.
1-4T5BVE 4.3	Fixed a server defect related to the threshold for reclaiming reserve memory.
1-6T0DTX 4.3	Fixed a server defect associated with XA commit of a transaction that included consuming a non-persistent message.
1-60KSHF 4.3	Fixed a server defect in which multiple receivers on an exclusive queue did not display correct fault-tolerant behavior.
1-6TMU6J 4.3	Fixed a server defect in which the server would swap a queue's messages into process storage, but would not swap them back out again. The trigger for this symptom involved a new queue consumer with a selector that did not match any messages.
1-6WJMLK 4.3	Fixed a server defect (HP-UX 11 only) in which large message selectors exhausted the storage stack.
1-6V6S5Q 4.3'	Fixed a server trace defect triggered when the value of a consumer `connectionID` was larger than 32 bits and `PRODCONS` tracing was enabled.

Reference # and Release	Description
1-40LEM5 4.3	Fixed a server trace defect in which connect failure trace did not honor `trace_client_host`.
1-6MVVLT 4.3	Improved server tracing when the attributes of a durable subscriber change.
1-6NKGQ4 4.3	Fixed a fault tolerance defect in which a server that was restarting as an FT standby could exit during the initialization protocol.
1-6OOLA0 4.3	Fixed a fault-tolerance defect related to misconfiguration of `max_connection`, which caused the server to exit soon after failover.
1-6OMN0J 4.3	Fixed a fault-tolerance defect triggered when a client application restarted during server failover.
1-6CMF6Z 4.3	Fixed a library defect in which servers and clients did not import CA certificates from PKCS12 files.
1-66ZS73 4.3	Fixed an LDAP parser defect affecting user names and group names containing special characters.
1-5JAQFL 4.3	Fixed an API defect in which calls to get SSL parameters erroneously returned FT SSL parameters.
1-6NC1JH 4.3	Fixed a client API defect in which load balanced connection factories would fail to create a connection if some of the listed servers were unavailable.
1-6K1XH4 4.3	Fixed a client API defect in which XA clients to fault-tolerant servers could lose messages if failover occurred before `xa_end`.
1-6JAOWV 4.3	Fixed a client API defect in which SSL connection factories did not properly import trusted certificates.
1-6OTQI3 4.3	Fixed a .NET client API defect involving transaction semantics across fault tolerance failover. Even though a `Session.Commit()` call threw a `TransactionRolledback` exception, the call could still erroneously acknowledge a message within the failed transaction.

Reference # and Release	Description
1-6RGO67 4.3	Fixed a .NET API defect in which `Tibems.GetProperty` did not return default property values when values were not explicitly set.
1-6W8163 4.3	Fixed a .NET API defect in which `QueueBrowser.MoveNext` behaved incorrectly at the end of a queue.
1-6VAI22 4.3	Fixed a .NET API defect in which `QueueBrowser.Reset` behaved incorrectly at the end of a queue.
1-6P4H9Z 4.3	Fixed a Java API defect in which clients supplying certificate authentication could not connect to a primary server that was not first in the list of FT servers.
1-5V1I1C 4.3	Fixed a Java API defect in which the method `createDurableConnectionConsumer` created non-durable topic subscribers.
1-5UKCE0 4.3	Fixed a Java API defect in which creating a `ConnectionConsumer` from a generic `Connection` object resulted in `ClassCastException`.
1-5F7IID 4.3	Fixed a Java API defect in which clients could not deserialize an array of custom class objects sent in an `ObjectMessage`.
1-6PKJ3D 4.3	Fixed a Java API defect in which creating a `ConnectionConsumer` on an `XAConnection` erroneously created a non-XA session.
1-6JAOXA 4.3	Fixed a Java API deficit by adding 3 forms of the method `setSSLIssuerCertificate`.
1-6MTG9D 4.3	Fixed a Java API defect in which the method `ServerInfo.getSSLParams` erroneously returned null.
1-1JKWUJ 4.3	Fixed a Java API defect related to parsing comma-separated lists of server URLs.
1-6VH87L 4.3	Fixed a Java API defect that affected sending messages that contained fields of type XML or that involved numeric conversions.

Reference # and Release	Description
1-6USUC5 4.3	Fixed a Java Admin API defect in which `ProducerInfo.toString` erroneously reported the connection ID, giving the value of the session ID instead.
1-6NBNI6 4.3	Fixed a C API defect in which creating more than one producer in a session could result in a memory leak.
1-6T5PF8 4.3	Fixed a C API defect in which closing a session without closing its consumers could result in a memory leak.
1-6S7R4U 4.3	Fixed a C API defect in which `tibemsQueueBrowser_GetNext` could erroneously retrieve messages more than once.
1-6NP9HV 4.3	Fixed a C API defect in which `tibemsSession_GetAcknowledgeMode` returned an incorrect value for a session with transaction semantics.
1-6T3IM3 4.3	Fixed a C API defect in which XA connection factories were incorrectly retrieved from an LDAP.
1-6W3VZ9 4.3	Fixed a C API defect in which client programs that used acknowledgement modes permitting duplicate delivery (`*_DUPS_OK_*`) would stop after fault-tolerant failover.
1-4Y2VET 4.3	Fixed a C API defect affecting stream messages containing null fields.
1-6NAQOH 4.3	Two previously deprecated C API calls are now obsolete. For details, see Obsolete on page 37.
1-6FIE9U 4.3	Fixed an administrative API defect in which the status of routes was incorrectly reported.
1-6FCQWQ 1-6FCQVX 4.3	Fixed an interaction defect with TIBCO Administrator, in which illegal user names were erroneously accepted, and subsequent unexpected behavior.
1-3FU5U5 4.3	Fixed an EMS administration tool defect in which the command `show connections type=s` did not display `Uptime` information.

Reference # and Release	Description
Issues Closed in Release 4.2	
1-4C6TK6 4.2	Fixed a defect importing Rendezvous messages to EMS, which affected the expiration topic property.
1-18QCAC 4.2	Fixed a defect in which client programs could not acknowledge a message after closing the consumer object (contrary to the JMS specification).
1-2CH7BB 4.2	Fixed a defect in which Java clients deadlocked during failover of a fault-tolerant EMS server.
1-2GWZC9 4.2	Fixed a server defect in which a fault-tolerant server pair did not properly failover to the secondary server.
1-2IPK9D 4.2	Fixed a server defect in which authorization interfered with fault-tolerant failover and routes.
1-2F47BS 4.2	Fixed a defect in which the server unnecessarily updated the files `queues.conf` and `topics.conf.`
1-2EQF3N 4.2	Fixed a defect in which in the administration tool would process and display uptime information incorrectly.
1-1ZG0YR 4.2	Fixed a server defect associated with the Linux arena memory allocation scheme. We have reduced thread contention for allocation calls, which improves the pattern of memory usage.
1-299SPD 4.2	When message tracking is disabled, the server no longer stores copies of message IDs. This change reduces memory usage and improves performance.
1-2AG3E9 4.2	Fixed a server defect associated with client authorizations. Generic `Session` objects (with proper authorization) can now create `QueueBrowser` objects.
1-19RISY 4.2	Fixed a defect in which the C client library did not properly detect duplicate delivery of persistent messages after fault tolerance failover.
1-1DI18A 4.2	Fixed a defect in the Java and C client APIs, in which closing a consumer and unsubscribing a durable in rapid sequence caused an error.

Reference # and Release	Description
1-1MLWPP 4.2	Fixed a defect in the EMS Hawk microagent, in which commands `purge topic`, `purge queue` and `purge durable` would improperly succeed, even when their arguments did not exist in the server.
1-1NB925 4.2	Fixed a memory leak in the EMS server during the transition from fault-tolerant standby mode to active mode.
1-1PISK9 4.2	Fixed a defect in which fault tolerance heartbeats were blocked by I/O difficulties.
1-1S8LPD 4.2	Fixed a race condition associated with destroying consumer objects.
1-1SSY7A 4.2	Fixed a defect associated with durable subscribers and the `nolocal` property.
1-1V8IRW 4.2	Fixed incorrect message swapping behavior of queue browsers.
1-1WQ6OO 4.2	Fixed a defect in which the server attempted to delete messages that it could not locate.
1-1XFRSC 4.2	Fixed a defect associated with the interaction between message compression (`JMS_TIBCO_COMPRESS`) and the `sender_name_enforced` property.
1-1YAWQN 4.2	Fixed a server defect associated with fault tolerance and client tracing, triggered by the administration tool command `set server client_trace=enabled`.
1-1YPX5Z 4.2	Fixed a defect in which name length limits were not enforced. Client ID names are now limited to 255 bytes. Durable names are now limited to 255 bytes.
1-1ZZVYH 4.2	Fixed a defect in which the fault tolerant standby server attempted to become active too soon during its start sequence.
1-25HC0D 4.2	Fixed a defect in which connection factories ignored `connect_attempt_count` when they specified only one server URL.

Reference # and Release	Description
1-25UGE3 4.2	Fixed a defect associated with two servers with reciprocal routes that are started simultaneously. This situation would produce this error message: `Invalid session for route configuration`
1-267VDH 4.2	Fixed a defect associated with the parameter `ft_reconnect_timeout`.
1-26DS49 4.2	Fixed a memory leak in the C function `tibemsLookupContext_Lookup`.
1-2CVD21 4.2	Fixed a defect in which the Java administration API created XA connection factories but did not correctly set the XA attribute.
1-2CVD2F 4.2	Fixed a defect associated with deleting a temporary queue that had a queue browser (that is, the browser is already closed). This action erroneously produced the following error message: `Attempt to delete temporary destination which has receivers`
1-2D8FNR 4.2	Fixed a client API defect associated with fault-tolerant failover.
1-2EICWI 4.2	Fixed a client API defect associated with the message property `JMSXDeliveryCount`.
1-2M3WNT 4.2	Fixed a defect in which the parameters `client_heartbeat` and `client_connection_timeout` interfered with fault-tolerant server reconnections.
1-2NYU5H 4.2	Fixed a defect in which connection factories did not use load balancing unless a load balancing metric was also set. That is, they did not use a default metric, but instead disabled load balancing.
1-2O50LT 4.2	Fixed a defect in the Java administration API method `updateFactory`, in which it did not send updated connect and reconnect values to the EMS server.
1-2O50M6 4.2	Fixed a defect in the Java administration API class `ConnectionFactoryInfo`; values for load balancing metrics were incorrect.
1-2QGIHI 4.2	Fixed a defect in which the server did not correctly propagate the administrative override value for the expiration property from destinations to messages.

Reference # and Release	Description
1-2SEA8X 1-318HQC 4.2	Fixed a defect associated with closing a session or a connection within an exception handler callback.
1-2TZ906 4.2	Fixed a defect in which accessing a fault-tolerant standby server from the administration tool or API could cause the server to exit.
1-318TAZ 4.2	Added a new C API entry point, `tibems_close()`, which Windows programs must call before unloading the EMS DLL.
1-32TS1L 4.2	Fixed a mislabeling of output from the administration tool `show routes` method.
1-33MCX5 4.2	Fixed the state name for FT standby state in the administration APIs (.NET and Java) `ServerInfo` class.
1-33MCYB 4.2	Fixed a defect associated with fault-tolerant connections, in which the EMS server recovered a consumer object without setting the connection that had created it. This defect prevented subsequent destruction of the consumer.
1-367ZMT 4.2	Enhanced the C API to automatically trim extraneous whitespace from URLs.
1-36RUOR 4.2	Fixed a client API defect in the method `Tibems.SetProperty()`.
1-371M1K 4.2	Fixed a defect in the EMS Hawk microagent. When using JRE 1.3, you must edit the `.hma` file to uncomment commands to include `jaxp.jar` and `crimson.jar`.
1-39QYL5 4.2	Fixed a server defect in which invalid destination names caused the server to exit.
1-3CIKIE 4.2	Fixed a server defect in which user authentication interfered with routing.
1-3FBOER 4.2	Fixed a server routing defect that resulted in this error message: `SEVERE ERROR: Received unexpected message type 55`

Reference # and Release	Description
1-3FX2GH 4.2	Fixed a server defect on Windows platforms, in which setting the `store_minimum_sync` parameter caused exit at start time.
1-3HWD4V 4.2	Fixed a defect in which the server would abnormally exit if LDAP authentication of a client returned failure after the client had already disconnected.
1-3HZ87L 4.2	Fixed a C API defect associated with the server's flow control feature. Programs would abnormally exit after this sequence of events: 1. Create a producer without a default destination. 2. Send non-persistent messages (to a server with flow control enabled). 3. Close the producer.
1-3MINZJ 4.2	Fixed a defect in which wildcard monitor topics could cause the server to terminate its client connection.
1-3VFKGF 4.2	Fixed a defect in which a fault-tolerant standby server would exhaust message memory.
1-3VNSYB 4.2	Fixed a server defect associated with routing, displaying two symptoms (in sequence). First, if the owner server stopped, the routed server would erroneously deliver messages sent directly to the routed queue. Subsequently, the restarted owner server would not route messages to the routed server.
1-44E74Z 4.2	Fixed a defect in which routed queues could erroneously be specified as exclusive (which is illegal).
1-46RMAH 4.2	Fixed a server defect in which a bridge would erroneously propagate the administrative override value for the expiration property to the target of the bridge.
1-4ACJ1L 4.2	Fixed a client API defect in which the message producer send methods did not clear the message's message ID header if message IDs were disabled.
1-4AVRES 4.2	Fixed a server defect in which LDAP records that did not match the configuration parameter `ldap_user_attribute` would cause the server to exit abnormally.
1-4ES278 4.2	Fixed a defect in which EMS supplied an incorrect URL for the administered object DTD.

Reference # and Release	Description
1-4GJPB7 4.2	Fixed a defect in which the server would write empty configuration files. this situation could occur when the disk partition was near full, and the server updated configuration files.
1-4IYSV1 1-5734ZE 4.2	Fixed a server defect in which the server could deliver duplicate messages after fault-tolerant failover.
1-4OW3GD 4.2	Fixed a defect in which setting `client_connection_timeout` or `server_connection_timeout` could cause the server to erroneously disconnect routes or clients.
1-4OW3GU 4.2	Fixed a defect in which the server misinterpreted the value of `client_heartbeat` as milliseconds instead of seconds.
1-4PGKQH 4.2	Fixed a defect in which the C function `tibemsMsg_Print` would erroneously print the names of JMS headers with the prefix `EMS`.
1-4SPY58 4.2	Fixed a server defect that associated with exporting messages to SmartSockets. The server did not export messages that were devoid of all JMS properties.
1-4T5BUL 4.2	Fixed an administration tool defect in which `show db` sometimes reported negative message sizes.
1-4TV376 4.2	Fixed a C client library defect in which send calls could overflow the message buffer, resulting in abnormal exit.
1-4U540B 4.2	Fixed a defect correlated with IBM WebSphere. Stopping and restarting the WebSphere Message Listener Port could produce the following exception: `javax.jms.IllegalStateException: Attempt to acknowledge messages which have not been sent to the client`
1-501ZJL 4.2	Fixed an administration API defect associated with promoting a passive route to an active route.
1-50HMQA 4.2	Fixed a client API defect that affected producers when all three of the following indicators were present—fault-tolerant servers, destination bridging and consumers with acknowledge mode `NO_ACKNOWLEDGE`.

Reference # and Release	Description
1-59Y969 4.2	Fixed an administration tool defect in the `autocommit` feature.
1-5AML2K 4.2	Fixed a server defect in which it was possible to create a `jndiname` on a dynamic destination (this is illegal).
1-5BKF9K 4.2	Fixed a server defect where commit operations could succeed when the transaction failed to process producer messages due to limits on topics or queues.
1-5D5VDK 4.2	Fixed a defect in which XA clients with fault-tolerant servers would fail when attempting to commit or roll back a transaction if the transaction's prepare state was not persistently stored in the server.
1-5BGSNE 4.2	Fixed a server defect in which a routed server could deliver duplicate messages to a consumer after the consumer had acknowledged them.
1-367ZMG 4.2	Fixed a defect in which sending non-persistent messages inappropriately fails in conjunction with flow control and overflowing message memory.
1-4X4KMX 4.2	Fixed a server defect in which malformed configuration parameters were silently ignored. The server now logs them.
Issues Closed in Release 4.1.0	
1-23LDOD 4.1.0	Fixed an API library defect in which recreated sessions within recreated connections did not properly acknowledge messages. This problem occurred only in the Java and .NET APIs, and when the acknowledge mode was `DUPS_OK`. In some cases, unbounded memory growth could occur as a secondary symptom.
1-21O7SO 4.1.0	Fixed a C API library defect in which acknowledge calls did not return. This problem could occur in sessions with either `CLIENT_ACKNOWLEDGE` or `EXPLICIT_CLIENT_ACKNOWLEDGE` mode, in situations where a program resent a message before acknowledging it.
1-21O7SG 4.1.0	Fixed a defect in which client API libraries did not clear the property `JMSRedelivered` before resending a message.
1-21JMZQ 4.1.0	Fixed a defect in which the server did not correctly set the property `JMSRedelivered` in recovered messages after server restart or fault-tolerant failover.

Reference # and Release	Description
1-1ZUSH7 4.1.0	Fixed a server defect in which flow control features did not operate when messages within a transaction were sent to a bridged destination.
1-1WS9N1 4.1.0	Fixed a defect in which TIBCO Hawk reported two servers (microagents) with the same name and display name.
1-1SHB62 4.1.0	Fixed a defect associated with character encodings in the Java and .NET client libraries.
1-1MGZS9 4.1.0	Fixed a defect associated with parsing configuration parameters, in which the parser did not report some types of malformed arguments as errors.
1-1OIAAD 4.1.0	Fixed a defect in which exclusive receivers did not receive messages after fault-tolerant failover of the server.
1-1Q0RIW 4.1.0	Fixed a defect in which the server refused name lookup requests after exceeding its `max_msg_memory` limit.
1-1QQYU9 4.1.0	Fixed a defect associated with LDAP authentication. Improved handling of subtree structures and attributes, to enable LDAP binding to use the `dn` returned by LDAP search.
1-1RVOEA 4.1.0	Fixed a defect associated with routed queues and fault-tolerant servers. After failover, routed queues did not deliver messages properly.
1-1SHEG5 4.1.0	Added methods to `ConnectionFactory` (Java and C APIs) to set connect and reconnect parameters.
1-1SMQPI 4.1.0	Fixed a defect associated with server names that are longer than 52 characters. (The maximum is 64 characters.)
1-1T2OT2 4.1.0	Fixed an administration tool defect. After modifying a route, the tool erroneously reported that it had modified a factory.
1-1V3FMX 4.1.0	Fixed a defect in which the server did not report an unsupported configuration as an error—namely, a routed queue cannot be exclusive.

Reference # and Release	Description
1-1V1KQH 4.1.0	Fixed a server defect in which administratively removing a message did not delete it from the undelivered queue.
1-1V0HUH 4.1.0	Fixed a defect associated with server restart, in which the administration tool erroneously reported undelivered messages (in the queue `$sys.undelivered`).
1-1UXRWQ 4.1.0	Fixed a defect in which authorization interfered with creating durable subscribers.
1-1UC04H 4.1.0	Fixed an omission in the configuration file `tibemsd.conf` (as distributed in release 4.0.0). It now explains how to specify `durables.conf`.
1-1TTNS2 1-1TTNRO 4.1.0	Fixed a defect in which the parser incorrectly handled server names containing the dot (`.`) character when configuring routes and routed queues.
1-1YOYO9 4.1.0	Fixed a defect in which the parser erroneously permitted user and group names containing the colon (`:`) character.
1-1TTNQ4 4.1.0	Clarified a Java exception that triggers when clients supply conflicting certificate and key files.
1-1TF6F5 4.1.0	Fixed a defect associated with the C function `tibemsSSLParams_SetIdentity` when the certificate data did not include the issuer.
1-1T2OUM 4.1.0	Fixed a server defect associated with long destination names (longer than 126 characters).
1-1T2OTU 4.1.0	Fixed a C API defect in which the functions `tibemsLookupContext_LookupConnectionFactory` and `tibemsLookupContext_LookupDestination` erroneously reported `TIBEMS_INVALID_ARG` when called with correct arguments.
1-1XTQR5 4.1.0	Fixed a defect associated with request time-out in the administration tool.
1-1YVAJ1 4.1.0	Fixed a defect in the server's TIBCO Hawk microagent. The `getRoute` and `getRoutes` methods now correctly output the `connected` field.

Reference # and Release	Description
1-1XKSBE 4.1.0	Fixed a .NET defect associated with calling the `Connection.Close` method while inside an exception listener event handler (`OnException`) or an exception delegate (`EMSExceptionHandler`).
1-1X2MLH 4.1.0	Fixed a defect associated with XA transactions in which it was possible to consume a message more than once.
1-1WBFW2 1-1VVFIH 4.1.0	Fixed a defect in the Java and C APIs associated with `xarecover` and `xarollback` of transactions.
1-1W6KQL 4.1.0	Fixed a defect in the Java API associated with XA transactions and fault-tolerant server failover.
1-1V8IQQ 4.1.0	Fixed a defect associated with parsing message selectors, which affected .NET API calls.
1-1SYODI 4.1.0	Fixed a defect associated with message selector expressions that contained parentheses, which affected the server and C API calls.
1-1SN68W 4.1.0	Fixed a defect associated with fault-tolerant failover when the server could not obtain a required file lock. The server now tries repeatedly to obtain that lock (rather than only once).
1-1RNTY9 4.1.0	Fixed a defect in the C API associated with message selectors that reference the `JMSTimestamp` header.
1-1QPE1X 4.1.0	Fixed a defect on the AIX platform, in which the server erroneously reported that it had exceeded memory limits.
1-1QBUCR 4.1.0	Fixed a defect associated with name lookup when the server is using its reserve memory.
1-1PVLZ2 4.1.0	Fixed a defect in which the server erroneously created missing configuration files. The affected files were `users.conf` and `groups.conf`. It is erroneous to create them automatically when `startup_abort_list = config_files` (otherwise it is correct to create them automatically).

Reference # and Release	Description
1-1P0H4F 4.1.0	Fixed a defect in which the client libraries (in all languages) erroneously omitted the property `JMSXDeliveryCount` when an inbound message did not define any other properties.
1-1IBQCY 4.1.0	Fixed a server defect that prevented clients with fault-tolerant connections from reconnecting to the active server after a network disconnect.
Issues Closed in Release 4.0.0	
1-1NP3G2 4.0.0	Fixed a defect associated with fault tolerance, in which a broken connection to a running server did not properly raise an exception in the client.
1-1MMPQ5 4.0.0	Fixed a server defect in which queues that have reached their `maxbytes` limit might erroneously discard messages when the server restarted.
1-1LTENW 4.0.0	Fixed a server defect associated with the fault tolerance feature and unidentified producers (that is, producers that do not specify a destination). The following error message was a symptom: `ERROR: No destination information for producer.`
1-1LB99C 4.0.0	Fixed a server defect associated with allocating reserve memory during database recovery.
1-1L4033 4.0.0	Performance improvements when Java clients access compressed messages. Apparent improvement depends on message size.
1-1634QX 4.0.0	Fixed on-line documentation about SSL parameters in the class `TibjmsAdmin`.
1-16THR4 4.0.0	Fixed a server defect in which C and Java clients did not properly flush acknowledgements in some configurations.
1-18QC9U 4.0.0	Fixed a defect in which the administration tool and API reported an incorrect estimate of memory usage when the number of queued messages was large.
1-18QCA7 4.0.0	Fixed a defect in which the server lagged when a large number of messages expired within a short time period.

Reference # and Release	Description
1-18SQ25 4.0.0	Fixed a defect in which the server did not reject a command to export a queue (which is illegal).
1-19RITO 4.0.0	Fixed a defect in which the C client library would needlessly recreate connections to fault-tolerant servers.
1-1A67VD 4.0.0	Improved server recovery time for large datastore files.
1-1-1BRCUX 4.0.0	Optimized the protocol by which servers share durable subscriber interest.
1-1DI160 4.0.0	Fixed a server defect associated with SSL.
1-1FGZ3L 4.0.0	Improved error reporting for maxbyte limits.
1-1FWCGX 4.0.0	Fixed a defect associated with overly-strict enforcement of non-critical X.509 v3 key usage extensions.
1-1FXC3B 4.0.0	Improved documentation for the `maxbytes` parameter, as it pertains to topics.
1-1HC2U7 4.0.0	Fixed a server defect associated with LDAP configuration parameters.
1-1HXOGO 4.0.0	Improved error reporting for routes in the server and administration tool.
1-1IBEJ4 4.0.0	Improved error texts for status codes 120, 159, 160.
1-1IM42Y 4.0.0	Fixed a defect associated with sending messages after calling the C function `tibemsMsg_GetByteSize()`.

Reference # and Release	Description
1-WWUWJ 4.0.0	Fixed a defect in which the server incorrectly imported some Rendezvous messages—converting to `MapMessage` instead of `TextMessage`. The presence of `JMSProperties` or `JMSHeaders` no longer prevents conversion to a `TextMessage`.
1-Y403U 4.0.0	Fixed incorrect names of Hawk (Java) packages in Appendix B of the Users's Guide.

Known Issues

Identified in Release	Summary/Workaround
Known Issues as of Release 5.1.5	
5.1.5	**Summary** When running EMS in FIPS compliant mode, DSA certificates cannot be used.
	Workaround Use RSA certificates when running EMS in FIPS compliant mode.
5.1.5	**Summary** On HP-UX 11.11, neither the OpenLDAP libraries (static or dynamic) nor the EMS server support SSL connections to an LDAP server.
	Workaround Connect to the LDAP server using a non-SSL connection.
Known Issues as of Release 5.1.4	
5.1.4	**Summary** The OpenLDAP shared libraries for HP-UX 11.11 do not support SSL connections to an LDAP server. Non-SSL connections are supported.
	Workaround When SSL connections are required, use the OpenLDAP static libraries.
5.1.4 1-AC2L2T	**Summary** On HPUX platforms, authentication with LDAP can fail in some situations when the `ldap_operation_timeout` parameter is set.
	Workaround If failure occurs, do not use `ldap_operation_timeout`.
Known Issues as of Release 5.1.1	
5.1.1	**Summary** Documentation for the unshared state failover feature in the .NET client is not available through the standard TIBCO Enterprise Message Service documentation interface.
	Workaround To access the documentation, open the `N_TIBCO_EMS_UFO.htm` file located in:
	EMS_HOME`\doc\html\tib_ems_api_reference\api\dotNETUFO\html`
Known Issues as of Release 5.1	
5.1 1-94OBDW	**Summary** Some clients may encounter errors when receiving messages with large Correlation IDs.
	Workaround The JMS Correlation ID should be limited to 4 KB in size.

Identified in Release	Summary/Workaround
Known Issues as of Release 5.0	
5.0	**Summary** The SmartSockets bridge is not supported for the 64-bit EMS server on the following platforms: • `hpux111/hppa` • `hpux112/ia64` **Workaround** None.
5.0	**Summary** Support for multicast, database storage, and extensible security features is not provided for HP-UX 11.11. **Workaround** None.
5.0	**Summary** Setting the `logfile` parameter in the `tibemsd.conf` file to a file path containing a space causes the EMS server to create the log file in the wrong place, unless the file path is surrounded by double quotes. **Workaround** Enclose the file path in double quotation marks.
5.0	**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.
Known Issues as of Release 4.x	
4.4 1-19RIVF	**Summary** While confirming messages from a routed queue, if the daemon where queue exists is killed and restarted, some of the messages will be redelivered to the client. **Workaround** None.
4.2 1-5DU9WW	**Summary** InstallShield problems prevent uninstalling EMS from Linux 24gl23 Itanium platform. **Workaround** Uninstall using this command line (all on one line): `java -cp TIBCO_EMS_HOME/_uninst/uninstaller.jar run`

Identified in Release	Summary/Workaround
4.1 1-22ZRNM	**Summary** JSSE cannot read PKCS12 certificates generated by some versions of OpenSSL. **Workaround** Import the certificate into a web browser; then export the certificate to a new file with extension `.p12` (not `.pfx`).