

TIBCO Enterprise Message Service™

Release Notes

*Software Release 4.3
February 2006*

Important Information

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

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

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

TIB, TIBCO, Information Bus, The Power of Now, TIBCO ActiveEnterprise, TIBCO Adapter, TIBCO Hawk, TIBCO Rendezvous, TIBCO Enterprise, TIBCO Enterprise Message Service, and the TIBCO logo are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, J2EE, JMS and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

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

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

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

Copyright © 1999–2006 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Release Notes	1
New Features	2
Compatibility With Previous Versions	12
Changes in Functionality	13
Deprecated & Obsolete Features	18
Release 4.3	18
Name Changes in Release 4.....	19
Executable Components.....	19
Configuration Files	19
C API	19
Java API	20
C# API	20
Closed Issues	21
Known Issues	49

Release Notes

This document includes release notes for TIBCO Enterprise Message Service, Software Release 4.3.

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, page 2*
- *Compatibility With Previous Versions, page 12*
- *Changes in Functionality, page 13*
- *Deprecated & Obsolete Features, page 18*
- *Name Changes in Release 4, page 19*
- *Closed Issues, page 21*
- *Known Issues, page 49*

New Features

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

Reference # and Release	Feature
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: <ul style="list-style-type: none"> • <code>maxmsgs</code> limits the number of unconsumed messages in a queue. • <code>overflowPolicy</code> offers a more flexible response when a queue overflows its <code>maxmsgs</code> or <code>maxbytes</code> 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 <code>SSLProxy</code> 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 <p>Some applications require strong or encrypted authentication, but do not require message encryption. The configuration parameter <code>ssl_auth_only</code> specifies SSL only during the authentication phase; subsequent messages do not incur the overhead of encryption.</p>

Reference # and Release	Feature
4.2	SSL Enhancements <ul style="list-style-type: none"> • Support for new SSL vendor <code>j3se-default</code> • Support for additional cipher suites in Java
1-UV9FQ	Dynamic Bridges
4.2	Administrators can now create and destroy bridges without restarting the server; see the <code>tibemsadmin</code> commands <code>create bridge</code> and <code>delete bridge</code> . New administrator permissions <code>view-bridge</code> and <code>change-bridge</code> regulate this feature.
1-1IL54R	.NET Administration API
4.2	.NET programs can use this new API to administer the EMS server. This API is functionally identical to the Java Administration API.
1-19TJZE	Permission to Use Durables
4.2	The ACL can now distinguish between permission to use an existing durable, and permission to create and configure a durable.
1-1WABEB	Enhanced Durable Information
4.2	Enhanced output from the administration tool command <code>show durable</code> (and the equivalent calls in the administration APIs). This information now includes the number of messages delivered to a durable.
1-22I3WV	Bridge Permission
4.2	Bridges have automatic permission to send to their target destinations.
1-1SEXX9	Enhanced Client Trace
4.2	Administrators can now set the <code>client_trace</code> parameter in the main configuration file (previous releases could set this parameter only using the administration tool).
1-3TYWXA	Undelivered SmartSockets GMD Messages
4.2	A new parameter, <code>preserve_gmd</code> , in <code>transports.conf</code> 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 <code>ServerInfo</code> class lets you get the message memory usage information from a server.
1-266AOV 4.2	EMS adds support for these AES cipher suites: <ul style="list-style-type: none"> • <code>TLS_RSA_WITH_AES_128_CBC_SHA</code> • <code>TLS_RSA_WITH_AES_256_CBC_SHA</code>
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, <code>-encryptedPassword</code> , which accepts a password encrypted with the Hawk utility program <code>tibhawkpassword</code> . This feature lets you store an encrypted password in the <code>.hma</code> 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 <code>/</code> qualifier replaces the existing cipher list, overriding existing <code>!</code> prefixes (which disable their ciphers).
1-519X6I 4.2	Enhanced administration tools and APIs to extract <code>user_auth</code> 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, <i>C & COBOL Reference</i> , replaces the old <i>C Reference</i> 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 <code>getServerInfo</code> . This method now includes more information about the server, such as message memory usage and process id.
1-1VYVEJ 4.1.0	New <code>-timeout</code> 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— <i>TIBCO Enterprise Message Service .NET API Reference</i> .
1-1G0DUF 4.1.0	.NET API now supports <code>TopicRequestor</code> and <code>QueueRequestor</code> 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 <code>Expiration</code> 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 <code>emsntsreg</code> registers <code>tibemsd</code> 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 <code>show server</code> 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	<p>.NET API enhancements:</p> <ul style="list-style-type: none"> 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	<p>When translating imported messages from TIBCO Rendezvous, fields of type <code>TIBRVMSG_XML</code> now translate to byte arrays in EMS messages. (In earlier releases, they erroneously translated to strings.)</p> <p>For backward compatibility with client programs that process XML fields as strings, the configuration parameter <code>tibrv_xml_import_as_string</code> restores the previous behavior). Nonetheless, we strongly recommend coding all new application programs to process XML fields as byte arrays.</p>
1-18QC8G 4.0.0	TIBCO SmartSockets interoperability. A new external transport type let you create a bridge between <code>tibemsd</code> and SmartSockets RTserver.

Reference # and Release	Feature
1-18T9EV 4.0.0	<p>Four new connection factory properties configure attempts to connect and reconnect to servers:</p> <ul style="list-style-type: none"> • <code>connect_attempt_count</code> • <code>connect_attempt_delay</code> • <code>reconnect_attempt_count</code> • <code>reconnect_attempt_delay</code>
1-1DADL2 4.0.0	<p>Four new server parameters configure heartbeat and timeout intervals for connections to clients and to other servers:</p> <ul style="list-style-type: none"> • <code>client_heartbeat</code> • <code>client_connection_timeout</code> • <code>server_heartbeat</code> • <code>server_connection_timeout</code>
1-14SG4O 4.0.0	<p>Documentation for server error messages now appears in an appendix to <i>TIBCO Enterprise Message Service User's Guide</i>.</p>
1-18QCAM 4.0.0	<p>Administrators can compact server database files to reclaim unused disk storage.</p>
1-18QC87 4.0.0	<p>Routing among servers allows both 1-hop and multi-hop zones, for greater control.</p>
1-18T9FH 4.0.0	<p>You can configure connection factories for load balancing among servers.</p>
1-UV9FA 4.0.0	<p><code>trace_client_host</code> is a new parameter in <code>tibemsd.conf</code>, which determines the way in which trace statements identify client hosts.</p>
1-VLPEQ 4.0.0	<p>Servers support certificate revocation lists (CRLs). Two new parameters in <code>tibemsa.conf</code> affect this feature—<code>ssl_crl_path</code> and <code>ssl_crl_update_interval</code>. A new administration tool command, <code>updatecrl</code>, immediately updates a server's CRL.</p>

Reference # and Release	Feature
1-19TJYZ 4.0.0	A new configuration file, <code>durables.conf</code> , allows static configuration of durable subscribers. This feature helps avoid message loss when starting a large system.
1-18QC9M 4.0.0	A new property of destinations lets administrators override the expiration property of messages.
1-1859TE 4.0.0	<code>startup_abort_list</code> is a new parameter in <code>tibems.conf</code> . 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: <ul style="list-style-type: none"> • <code>ldap_conn_type</code> • <code>ldap_tls_cacert_file</code> • <code>ldap_tls_cacert_dir</code> • <code>ldap_tls_ciphers</code> • <code>ldap_tls_rand_file</code> • <code>ldap_tls_cert_file</code> • <code>ldap_tls_key_file</code>

New Features in Release 3.1.2

- 3.1.2 The C client library adds two functions to set the size of connection buffers:
- `tibems_SetSocketSendBufferSize()`
 - `tibems_SetSocketReceiveBufferSize()`

New Features in Release 3.1.1

- 3.1.1 This product now supports Itanium hardware (ia64) with these operating systems: HP-UX 11.x and RedHat Linux (2.4 kernel).
- 3.1.1 Added a chapter about WebLogic 8.1 integration to *TIBCO Enterprise Message Service Application Integration Guide*.
- 3.1.1 Added the server URL to the Hawk MicroAgent name.
- 3.1.1 Added an `isRunning()` method to the Hawk MicroAgent.

Reference # and Release	Feature
3.1.1	Java Administration API—added methods to get and set selectors on routes.
3.1.1	Improved server performance for queues with several consumers that filter messages using selectors.
New Features in Release 3.1.0	
3.1.0	<p>Fully-managed C# client for Microsoft .NET environment.</p> <p>This feature was previously a technology preview, but now it is an officially supported feature of TIBCO Enterprise Message Service.</p>
3.1.0	<p>Server-based destination bridging. This allows you to define a bridge between two or more destinations, possibly of different types, so that messages sent to one destination are also sent to the bridged destinations.</p> <p>This feature was previously a technology preview, but now it is an officially supported feature of TIBCO Enterprise Message Service.</p>
3.1.0	<p>Support for multiple "transports" to be defined for destinations. These transports supersedes the existing TIBCO Rendezvous bridge import/export functionality. Now transports are defined in a configuration file and are then used with the new <code>import</code> and <code>export</code> destination properties.</p> <p>The old mechanism for communicating with TIBCO Rendezvous is still fully functional and supported, but it is deprecated in favor of the new mechanism. You can continue to use the old mechanism, but it is strongly recommended that you update your configuration to use transports.</p>
3.1.0	Flow control for destinations so that message producers can be slowed down if they are producing messages faster than message consumers can receive them.
3.1.0	Support for external systems, such as an LDAP server, for user authentication and group information.
3.1.0	TIBCO Hawk MicroAgent for monitoring and managing the TIBCO Enterprise Message Service server.
3.1.0	<code>TibrvJMSTransport</code> class for TIBCO Rendezvous Java applications and applets to connect to the TIBCO Enterprise Message Service server.
3.1.0	Functions for handling character encodings for strings have been added to the APIs. Also, a character encoding conversion library has been included in the C client API.

Reference # and Release	Feature
3.1.0	Monitoring and tracing can now be performed on a per-message basis.
3.1.0	The C client API has been enhanced to include functions for looking up objects (such as ConnectionFactories and Destinations) from the TIBCO Enterprise Message Service server. This functionality is similar to using JNDI in a Java client to lookup objects.
3.1.0	Added support for <i>no prefetch</i> queues. Receivers on <i>no prefetch</i> queues do not receive messages in the background. Instead, they only receive the next message when calling one of the receive methods or when returning from a user's message callback.
3.1.0	Queues can specify a redelivery policy. You can specify the <code>maxRedelivery</code> property on the queue to determine the maximum number of times a message should be redelivered. When the server has reached the maximum number of delivery attempts, the message is destroyed or placed on the undelivered queue depending upon whether the <code>JMS_TIBCO_PRESERVE_UNDELIVERED</code> property is set in the message.
3.1.0	Support for AIX 5.1.
3.1.0	Additional examples are available for C# clients.
New Features in Release 3.0.1	
3.0.1	HP Tru64 Unix 5.1A is now supported.
New Features in Release 3.0.0	
3.0.0	Full support of JMS 1.1 specifications in Java and C clients.
3.0.0	Full XA support in Java and C clients.
3.0.0	General availability release of the C client.
3.0.0	Fine-grained administrative permissions.
3.0.0	Detailed run-time statistics for producers, consumers, destinations and routed servers.
3.0.0	Server monitoring facilities.
3.0.0	Server log file rotation by size or by administrative command.

Reference # and Release	Feature
3.0.0	Selector-based topic routing.
3.0.0	Support for external SSL accelerators.
3.0.0	Extended server tracing—added new trace items.
3.0.0	Support for message identification by inclusion of the sender name into the message.
3.0.0	Ability to dynamically pre-register or remove TIBCO Rendezvous Certified Messaging clients in the server.
3.0.0	Added JMS 1.1-based C and Java samples, XA samples.
3.0.0	Documentation The <i>TIBCO Enterprise Message Service User's Guide</i> has been rewritten, restructured, and improved. Also, a new book has been added that describes the C API, <i>TIBCO Enterprise Message Service C & COBOL API Reference</i> .

Compatibility With Previous Versions

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.

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 18.
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	<p>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:</p> <p>When a client closes a consumer with an unacknowledged message, the server no longer reinstate that message back onto its queue.</p> <p>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 <code>tibemsd.conf</code>:</p> <pre>compliant_consumer_close = disabled</pre> <p>Otherwise, we strongly recommend leaving this parameter as enabled (the default value).</p>
4.2	<p>SSL</p> <p>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 <code>entrust61</code> (it has not changed).</p>

Reference # and Release	Functionality
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, <code>tibems_close()</code> , which Windows programs must call before unloading the EMS DLL.
4.2	The C function <code>tibemsMsg_ByteSize</code> is deprecated. Use <code>tibemsMsg_GetByteSize</code> 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) <code>ServerInfo</code> class. Administration programs that used the old deprecated value (<code>SERVER_STANDBY</code>) require updating to the new value (<code>SERVER_FT_STANDBY</code>).
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 <code>Session.Run</code> is obsolete.

Reference # and Release	Functionality
4.1.0	<p>In the Java API, the following methods of class <code>Tibjms</code> are deprecated: <code>getAllowCallbackInClose</code> and <code>setAllowCallbackInClose</code>. These items were misnamed; we recommend eliminating references to them from your code, replacing them with references to <code>getAllowCloseInCallback</code> and <code>setAllowCloseInCallback</code> (respectively).</p> <p>In the .NET API, the corresponding methods of class <code>Tibems</code> are deprecated: <code>GetAllowCallbackInClose</code> and <code>SetAllowCallbackInClose</code>. We recommend eliminating references to them from your code, replacing them with references to <code>GetAllowCloseInCallback</code> and <code>SetAllowCloseInCallback</code> (respectively).</p>
1-1IQQZW 4.1.0	For new configuration details regarding JBoss 3.2.3, see <i>TIBCO Enterprise Message Service Application Integration Guide</i> .
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 19.
1-1L3HOR 4.0.0	<p>When translating imported messages from TIBCO Rendezvous, fields of type <code>TIBRVMSG_XML</code> now translate to byte arrays in EMS messages. (In earlier releases, they erroneously translated to strings.)</p> <p>For backward compatibility with client programs that process XML fields as strings, the configuration parameter <code>tibrv_xml_import_as_string</code> restores the previous behavior). Nonetheless, we strongly recommend coding all new application programs to process XML fields as byte arrays.</p>
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.
Release 3.1.2	
3.1.2	The server's <code>max_connections</code> parameter applies only to client connections. It no longer limits connections from other servers or from administrative programs.

Reference # and Release	Functionality
Release 3.1.1	
3.1.1	<p>In the C# client library, the <code>Lookup</code> method of the <code>LookupContext</code> throws different exceptions than in earlier releases.</p> <ul style="list-style-type: none"> When the method times out while waiting for a response from <code>tibjmsd</code>, it throws <code>ServiceUnavailableException</code> (previously <code>CommunicationException</code>). When the method cannot find the requested object, it throws <code>NameNotFoundException</code> (previously <code>CommunicationException</code>).
Release 3.1.0	
3.1.0	<p>Importing and exporting messages to and from TIBCO Rendezvous is now done with transports and the new <code>import</code> and <code>export</code> properties. The new functionality supersedes the old TIBCO Rendezvous and certified messaging <code>import</code> and <code>export</code> properties.</p> <p>The old configuration will still function properly, but you should move to the new method of configuring transports as soon as possible. The old destination properties and configuration parameters are deprecated and may be removed in a future version of TIBCO Enterprise Message Service.</p>
3.1.0	Queue receivers on a queue routed from another server are no longer allowed to specify <code>NO_ACKNOWLEDGE</code> mode.
3.1.0	<p>Entrust Version 6.0 is no longer supported as the SSL vendor for Java clients. You can no longer specify <code>entrust6</code> in the <code>TibjmsContext.SSL_VENDOR</code> property. If you wish to use Entrust, you must use Entrust Version 6.1. You can specify <code>entrust61</code> in the <code>TibjmsContext.SSL_VENDOR</code> property. See <i>TIBCO Enterprise Message Service User's Guide</i> and your Entrust documentation for more information about configuring your client to use the Entrust libraries.</p>
Release 3.0.1	
3.0.1	<p>Message delivery for non-exclusive queues has changed. Now, messages are delivered in a round-robin fashion to balance the load of incoming messages across all queue receivers. See the description of the <code>exclusive</code> property for queues in <i>TIBCO Enterprise Message Service User's Guide</i> for more information.</p>

Reference # and Release	Functionality
Release 3.0.0	
3.0.0	Changes to the C Client API <ul style="list-style-type: none">• Changed <code>tibjmsTopic</code> and <code>tibjmsQueue</code> to be typedef of <code>tibjmsDestination</code>. This does not affect existing programs.• Public constants <code>_TIBJMS_TOPIC_</code>, <code>_TIBJMS_QUEUE_</code>, and <code>_TIBJMS_UNKNOWN_</code> have been renamed to <code>TIBJMS_TOPIC</code>, <code>TIBJMS_QUEUE</code> and <code>TIBJMS_UNKNOWN</code>.
3.0.0	Improved server behavior in case of large message backlog resulting in extensive message swap out.

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 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
<code>tibemsStreamMsg_GetBytes</code>	<code>tibemsStreamMsg_ReadBytes</code>
<code>tibemsStreamMsg_SetBytes</code>	<code>tibemsStreamMsg_WriteBytes</code>

Obsolete

UNIX system password authentication is no longer supported.

Deprecated

SSL key renegotiation is deprecated in release 4.3; it is not supported in release 5.0.

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

- `tibemspd` 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

- `tibemspd.conf` replaces `tibjmsd.conf`.

The server first attempts to locate an existing file named `tibemspd.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 `tibemspd.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.

Closed Issues

This section lists issues that were closed in the named releases.

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: <ul style="list-style-type: none"> • 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 <i>both</i> 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 <code>meta.db</code> or <code>sync-msgs.db</code> 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 <code>ssl_auth_only</code> .
1-5VHKG1 4.3	Fixed a server defect in which the file <code>async-msgs.db</code> 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.

Reference # and Release	Description
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.
1-5JW4Q5 4.3	Fixed a server defect in which topics with transaction semantics did not respect destination limits, such as <code>maxBytes</code> .
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 <code>sender_name</code>) 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.

Reference # and Release	Description
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.
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.

Reference # and Release	Description
1-6JAOVV 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 <code>Session.Commit()</code> call threw a <code>TransactionRolledback</code> exception, the call could still erroneously acknowledge a message within the failed transaction.
1-6RGO67 4.3	Fixed a .NET API defect in which <code>Tibems.GetProperty</code> did not return default property values when values were not explicitly set.
1-6W8163 4.3	Fixed a .NET API defect in which <code>QueueBrowser.MoveNext</code> 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 <code>createDurableConnectionConsumer</code> created non-durable topic subscribers.
1-5UKCE0 4.3	Fixed a Java API defect in which creating a <code>ConnectionConsumer</code> from a generic <code>Connection</code> object resulted in <code>ClassCastException</code> .
1-5F7IID 4.3	Fixed a Java API defect in which clients could not deserialize an array of custom class objects sent in an <code>ObjectMessage</code> .
1-6PKJ3D 4.3	Fixed a Java API defect in which creating a <code>ConnectionConsumer</code> on an <code>XAConnection</code> erroneously created a non-XA session.
1-6JAOXA 4.3	Fixed a Java API deficit by adding 3 forms of the method <code>setSSLIssuerCertificate</code> .
1-6MTG9D 4.3	Fixed a Java API defect in which the method <code>ServerInfo.getSSLParams</code> erroneously returned null.
1-1JKWUJ 4.3	Fixed a Java API defect related to parsing comma-separated lists of server URLs.

Reference # and Release	Description
1-6VH87L 4.3	Fixed a Java API defect that affected sending messages that contained fields of type XML or that involved numeric conversions.
1-6NBN16 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 <code>tibemsQueueBrowser_GetNext</code> could erroneously retrieve messages more than once.
1-6NP9HV 4.3	Fixed a C API defect in which <code>tibemsSession_GetAcknowledgeMode</code> 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 (<code>*_DUPS_OK_*</code>) 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 18.
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 <code>show connections type=s</code> did not display <code>Uptime</code> 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 <code>queues.conf</code> and <code>topics.conf</code> .
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 <code>purge topic</code> , <code>purge queue</code> and <code>purge durable</code> 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 <code>noLocal</code> 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 (<code>JMS_TIBCO_COMPRESS</code>) and the <code>sender_name_enforced</code> property.
1-1YAWQN 4.2	Fixed a server defect associated with fault tolerance and client tracing, triggered by the administration tool command <code>set server client_trace=enabled</code> .
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 <code>connect_attempt_count</code> 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 <code>ft_reconnect_timeout</code> .
1-26DS49 4.2	Fixed a memory leak in the C function <code>tibemsLookupContext_Lookup</code> .
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 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 <code>JMSXDeliveryCount</code> .
1-2M3WNT 4.2	Fixed a defect in which the parameters <code>client_heartbeat</code> and <code>client_connection_timeout</code> 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 <code>updateFactory</code> , 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 <code>ConnectionFactoryInfo</code> ; values for load balancing metrics were incorrect.

Reference # and Release	Description
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.
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, <code>tibems_close()</code> , which Windows programs must call before unloading the EMS DLL.
1-32TS1L 4.2	Fixed a mislabeling of output from the administration tool <code>show routes</code> method.
1-33MCX5 4.2	Fixed the state name for FT standby state in the administration APIs (.NET and Java) <code>ServerInfo</code> 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 <code>Tibems.SetProperty()</code> .
1-371M1K 4.2	Fixed a defect in the EMS Hawk microagent. When using JRE 1.3, you must edit the <code>.hma</code> file to uncomment commands to include <code>jaxp.jar</code> and <code>crimson.jar</code> .
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.

Reference # and Release	Description
1-3FBOER 4.2	Fixed a server routing defect that resulted in this error message: SEVERE ERROR: Received unexpected message type 55
1-3FX2GH 4.2	Fixed a server defect on Windows platforms, in which setting the <code>store_minimum_sync</code> 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: <ol style="list-style-type: none"> 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.

Reference # and Release	Description
1-4AVRES 4.2	Fixed a server defect in which LDAP records that did not match the configuration parameter <code>ldap_user_attribute</code> 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.
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 <code>client_connection_timeout</code> or <code>server_connection_timeout</code> 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 <code>client_heartbeat</code> as milliseconds instead of seconds.
1-4PGKQH 4.2	Fixed a defect in which the C function <code>tibemsMsg_Print</code> would erroneously print the names of JMS headers with the prefix <code>EMS</code> .
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 <code>show db</code> sometimes reported negative message sizes.
1-4TV376 4.2	Fixed a C client library defect in which <code>send</code> 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: <pre>javax.jms.IllegalStateException: Attempt to acknowledge messages which have not been sent to the client</pre>

Reference # and Release	Description
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.
1-59Y969 4.2	Fixed an administration tool defect in the <code>autocommit</code> feature.
1-5AML2K 4.2	Fixed a server defect in which it was possible to create a <code>jndiname</code> 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 <code>DUPS_OK</code> . 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 <code>CLIENT_ACKNOWLEDGE</code> or <code>EXPLICIT_CLIENT_ACKNOWLEDGE</code> mode, in situations where a program resent a message before acknowledging it.

Reference # and Release	Description
1-21O7SG 4.1.0	Fixed a defect in which client API libraries did not clear the property <code>JMSRedelivered</code> before resending a message.
1-21JMZQ 4.1.0	Fixed a defect in which the server did not correctly set the property <code>JMSRedelivered</code> in recovered messages after server restart or fault-tolerant failover.
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 <code>max_msg_memory</code> 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 <code>dn</code> 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 <code>ConnectionFactory</code> (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.)

Reference # and Release	Description
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.
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 <code>\$sys.undelivered</code>).
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 <code>tibemsd.conf</code> (as distributed in release 4.0.0). It now explains how to specify <code>durables.conf</code> .
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 dot (.) 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 <code>tibemsSSLParams_SetIdentity</code> 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 <code>tibemsLookupContext_LookupConnectionFactory</code> and <code>tibemsLookupContext_LookupDestination</code> erroneously reported <code>TIBEMS_INVALID_ARG</code> when called with correct arguments.

Reference # and Release	Description
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 <code>getRoute</code> and <code>getRoutes</code> methods now correctly output the <code>connected</code> field.
1-1XKSBE 4.1.0	Fixed a .NET defect associated with calling the <code>Connection.Close</code> method while inside an exception listener event handler (<code>OnException</code>) or an exception delegate (<code>EMSEExceptionHandler</code>).
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 <code>xarecover</code> and <code>xarollback</code> 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 <code>JMSTimestamp</code> header.
1-1QP1X 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.

Reference # and Release	Description
1-1PVLZ2 4.1.0	Fixed a defect in which the server erroneously created missing configuration files. The affected files were <code>users.conf</code> and <code>groups.conf</code> . It is erroneous to create them automatically when <code>startup_abort_list = config_files</code> (otherwise it is correct to create them automatically).
1-1P0H4F 4.1.0	Fixed a defect in which the client libraries (in all languages) erroneously omitted the property <code>JMSXDeliveryCount</code> 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 <code>maxbytes</code> 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 <code>TibjmsAdmin</code> .
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.

Reference # and Release	Description
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.
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 <code>maxbytes</code> 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.

Reference # and Release	Description
1-IIM42Y 4.0.0	Fixed a defect associated with sending messages after calling the C function <code>tibemsMsg_GetByteSize()</code> .
1-WWUWJ 4.0.0	Fixed a defect in which the server incorrectly imported some Rendezvous messages—converting to <code>MapMessage</code> instead of <code>TextMessage</code> . The presence of <code>JMSProperties</code> or <code>JMSHeaders</code> no longer prevents conversion to a <code>TextMessage</code> .
1-Y403U 4.0.0	Fixed incorrect names of Hawk (Java) packages in Appendix B of the Users's Guide.
Issues Closed in Release 3.1.2	
3.1.2	Fixed a memory leak in the server, triggered by clients creating and destroying SSL connections in a tight loop.
3.1.2	Fixed a memory leak in the C client API associated with SSL.
3.1.2	Fixed a memory leak in the C API function <code>tibjmsMsg_CreateFromBytes()</code> .
3.1.2	Fixed a timing defect associated with closing <code>DUPS_OK</code> sessions. This action infrequently could cause the application program to abruptly exit.
3.1.2	Fixed a server defect associated with <code>max_msg_memory</code> . After a server reached that maximum, restarting the server would result in a severe error and discarded messages.
3.1.2	Fixed a defect in the C# client library, in which messages did not expire properly.
3.1.2	Fixed a defect in the C# client library. When a queue with <code>prefetch=none</code> already contained messages before a C# receiver starts, the receiver stopped after consuming the first message.
3.1.2	Fixed a defect in the client library, in which queue browsers did not detect messages in the undelivered message queue.
3.1.2	Fixed a memory leak in the client library, associated with deleting temporary topics.
3.1.2	Fixed a defect that affected clients using JNDI lookup to locate the server. When the server had exceeded its memory limit, JNDI lookup failed, reporting timeout.

Reference # and Release	Description
3.1.2	Fixed a defect associated with the destination attributes <code>sender_name</code> and <code>sender_name_enforced</code> . When these properties were set, Java client receivers would throw an exception.
3.1.2	Fixed a server defect associated with queues that have attributes <code>exclusive</code> and <code>prefetch=none</code> . Queue receivers would present only one message, even when the server had other messages in the queue.
3.1.2	Fixed a server memory leak associated with queue browsers. This defect also affected the message memory usage information (in the administration tool and API).
3.1.2	Removed a dependency of the C client library on additional Solaris libraries (Solaris platforms only).
3.1.2	Fixed a defect in which server ignored its <code>max_connections</code> parameter.
3.1.2	Fixed a defect in the server's Hawk microagent. Both the <i>name</i> and the display name of the microagent now include the server's URL (to differentiate among several servers).
3.1.2	Fixed a defect in the C client library associated with string properties of messages. If a client program received a message with string properties, added another string property, and sent the message again—then the send call would fail.
3.1.2	Fixed a memory leak in the client library, associated with closing <code>MsgRequestor</code> objects.
3.1.2	Fixed a defect in the administrative interfaces, in which inherited destination attributes <code>sender_name</code> and <code>sender_name_enforced</code> were marked as + (set directly on the destination) rather than * (inherited).
3.1.2	Fixed a defect on SMP hardware. When a queue in <code>CLIENT_ACKNOWLEDGE</code> mode delivered a message to a multi-threaded client, and the client consumed the message in one thread, and acknowledged it in a second thread, then the queue erroneously retained the message as unacknowledged.

Issues Closed in Release 3.1.1

3.1.1	Fixed a defect in C API library. When a producer program (in C) sent a set of non-persistent messages to a server and immediately closed its connection, the connection did not correctly flush all the messages to the server; consequently, the unflushed messages were lost.
-------	---

Reference # and Release	Description
3.1.1	<p>Fixed a defect in the server regarding permissions for administrative commands. The following commands require <code>view-server</code> permission (rather than <code>change-server</code>):</p> <pre> show rvcmlledger show rvcmtransportledger show rvcmllisteners show transports show transport show bridges show bridge </pre>
3.1.1	<p>Fixed a defect in the implementation of XA within the server, and the Java and C client libraries. The following two valid call sequences did not complete properly:</p> <pre> xa_start xid xa_end xid SUSPEND xa_end xid SUCCESS xa_start xid xa_end xid SUSPEND xa_end xid FAIL </pre>
3.1.1	<p>Fixed a defect in which the server did not preserve undeliverable messages after termination and restart. This symptom would affect messages for which the server exceeded <code>maxRedelivery</code> attempts, and consequently had moved the message to the system queue <code>#sys.undelivered</code>.</p>
3.1.1	<p>Fixed a defect in which a restarted server might erroneously redeliver <code>NON_PERSISTENT</code> messages which a durable consumer had acknowledged (before server restart).</p>
3.1.1	<p>Fixed a defect in which routed queues might stop delivering messages to consumers, or might erroneously redeliver messages to consumers. These symptoms were limited to situations in which an application closed one or more consumers of the routed queue, leaving other consumers active.</p>
3.1.1	<p>Fixed a defect in which a global queue could deliver duplicate messages. The problem affected global queues with more than one remote consumer, after one of those consumers closed its connection.</p>
3.1.1	<p>Fixed a defect that affected a server with a destination bridge from a topic to a queue. If the server exceeded available memory, and consequently released its reserve memory, the bridge could lose messages.</p>

Reference # and Release	Description
3.1.1	Fixed a defect in which the C sample program <code>tibjmsXAMsgProducer</code> with command line parameter <code>-help</code> produced a segmentation fault on Solaris 7 platforms.
3.1.1	Fixed a defect in the C client library. When the client lost its connection to <code>tibjmsd</code> , its <code>receive</code> call would unblock before the exception callback completed. C client behavior in this situation is now identical to Java client behavior. However, it is now illegal to call <code>tibjmsConnection_Close()</code> while an <code>onException()</code> callback is running (in any thread); attempting to do so results in the error code <code>TIBJMS_ILLEGAL_STATE</code> .
3.1.1	For C clients, calling <code>tibjmsConnection_Stop()</code> inside the <code>exceptionListener</code> callback no longer causes problems.
3.1.1	Fixed a defect in which the method <code>com.tibco.tibjms.admin.ServerInfo.getFaultTolerantURL</code> and the <code>tibjmsadmin</code> command <code>show config</code> would erroneously report that the configuration parameter <code>ft_active</code> was <code>null</code> , when the server was active.
3.1.1	Fixed a defect in which a consumer application could leave unacknowledged messages at the daemon. This symptom occurred when a consumer application used <code>DUPS_OK</code> or <code>EXPLICIT_CLIENT_DUPS_OK</code> as the acknowledge mode, and closed a consumer object before closing its connection or session objects.
3.1.1	Fixed a defect in which a C client that created and destroyed connections in a tight loop could cause Bad File Descriptor errors in <code>select</code> calls.
3.1.1	Fixed a defect in the <code>purgeDurable</code> method of the class <code>com.tibco.tibjms.admin.HawkController</code> . The client ID argument is optional, and the method now accepts a <code>null</code> value as an omitted argument.
3.1.1	Fixed a defect with non-XA transactions. When a program closed a connection before closing all of its sessions, the transaction records remained in the store file; the server would remove them the next time it restarted.
3.1.1	Fixed a defect in the <code>getOutboundBytesRate</code> method of the class <code>com.tibco.tibjms.admin.ServerInfo</code> . It erroneously returned the outbound message rate rather than the outbound byte rate.
3.1.1	Java clients now run properly with JDK 1.2.1.

Reference # and Release	Description
3.1.1	<p>For C clients, memory leaks have been corrected in these situations:</p> <ul style="list-style-type: none"> • sending compressed messages • closing a connection before closing the session • closing a session with durable subscribers • <code>tibjmsMsg_GetAsBytes</code> and <code>tibjmsMsg_GetAsBytesCopy</code>
3.1.1	Fixed a defect that prevented setting <code>flowControl</code> on destinations with limit size greater than or equal to 4GB.
3.1.1	Fixed a defect in setting the <code>maxBytes</code> property in the configuration files <code>topics.conf</code> and <code>queues.conf</code> , in which values larger than 4GB were truncated to their low-order 32 bits.
3.1.1	Fixed a defect in which exporting messages to TIBCO Rendezvous would fail when the <code>replyTo</code> field of the exported message was a temporary destination.
3.1.1	For .NET clients, fixed a defect which caused client exceptions when receiving messages.
3.1.1	Fixed a defect in which .NET clients incorrectly reported <code>Malformed server response</code> when a JNDI request to the server could not locate a requested object.
3.1.1	Fixed a defect in which Java clients and .NET clients incorrectly reported <code>Malformed server response</code> when a JNDI request to the server exceeded its time limit.
3.1.1	Fixed a defect in which Java clients sending JNDI requests would timeout when the server reached its message memory limit.
3.1.1	Fixed a memory leak in the server, which was caused by incorrect route configuration.
3.1.1	Fixed a memory leak in the server when clients connect to it using SSL.
3.1.1	Fixed a defect in the administration API, in which the constructor <code>ConnectionFactoryInfo()</code> ignored the <code>xa</code> flag.
3.1.1	For C clients, fixed a defect when republishing a message received in an asynchronous callback. The symptoms were memory leak and unacknowledged messages (in <code>AUTO_ACKNOWLEDGE</code> mode).

Reference # and Release	Description
3.1.1	Fixed a defect in which the server created redundant durable subscribers for a route when incoming topic selectors were set.
3.1.1	Fixed a defect in which the server ignored incoming topic selectors for a route while the route was off-line.
3.1.1	Fixed a defect in which a restarted server erroneously sent topic messages back to the sending server.
3.1.1	Fixed a defect in which a restarted server erroneously delivered local topic messages to subscribers that had specified <code>noLocal</code> .
3.1.1	Fixed a defect in which a server erroneously delivered old messages after a durable subscriber changed the value of <code>noLocal</code> .
3.1.1	Fixed a defect in which JNDI aliases were missing from <code>TopicInfo</code> and <code>QueueInfo</code> objects when returned by <code>getTopics</code> and <code>getQueues</code> methods.
3.1.1	Deprecated methods related to old-style RV & RVCN transport configuration are now marked as deprecated in the JavaDoc.
3.1.1	Fixed a flow control defect in which producers could stall when messages were removed from the destination into an undelivered queue.
3.1.1	The length of the selector string in server configuration files was limited to 2000 characters. We have increased it to allow selectors up to 30,000 characters long.
3.1.1	Fixed a message-corrupting defect that affected compressed messages in which the <code>trace</code> property was set.
3.1.1	Java samples now correctly handle the SSL parameters.
Issues Closed in Release 3.1.0	
3.1.0	For C clients, calling <code>tibjmsSession_Rollback()</code> on a non-transacted session now correctly returns <code>TIBJMS_ILLEGAL_STATE</code> instead of <code>TIBJMS_INVALID_SESSION</code> .
3.1.0	For C clients, not all messages on a transacted session were delivered after a rollback. This has been fixed.
3.1.0	For C clients, duplicate messages were occasionally sent to a consumer after a session was recovered. This has been fixed.

Reference # and Release	Description
3.1.0	For C clients, a crash occurred after <code>tibjmsMsg_SetJMSType()</code> was called on a message that was just received. This has been fixed.
3.1.0	For C clients, consumers using <code>TIBJMS_NO_ACKNOWLEDGE</code> mode had a memory leak. This has been fixed.
3.1.0	For C clients, calling <code>tibjmsTextMsg_SetText()</code> or <code>tibjmsObjectMsg_SetObjectBytes()</code> more than once caused a memory leak. This has been fixed.
3.1.0	Java clients would occasionally fail with the message "TCPLink Error: invalid magic in the message". This has been fixed.
3.1.0	<code>TibjmsConnectionConsumer</code> now loads the specified number of messages into the <code>Session</code> before calling the <code>start</code> method on the <code>ServerSession</code> .
3.1.0	Delivery of messages on a routed queue was stalled when a client performed XA rollbacks. This has been fixed.
3.1.0	When a <code>show</code> command is issued in the administration tool, an error is no longer returned if there are no items to show.
3.1.0	In a Java client, if the flag passed to the <code>XAResource</code> 's <code>recover</code> method was anything other than <code>TMSTARTSCAN</code> , <code>recover</code> returned a null instead of an array of zero length. This has been fixed.
3.1.0	After a topic or queue was destroyed then recreated, RVCML messages were no longer imported. This has been fixed.
3.1.0	When a client failed to write data into a message in a transacted session, the transaction was not always failing. This has been fixed.
3.1.0	The server returned an immediate response to a topic publisher even though any messages sent over a bridge were still being processed. Also, an IO failure on a message sent over a bridge was not causing an error response to the original sender. These defects have been fixed.
3.1.0	The <code>getRVCMLedger</code> method is now added to the <code>TibjmsAdmin</code> class to provide the functionality of the <code>show rvcmlledger</code> command in the administration tool.
3.1.0	Messages committed as part of a local or an XA transaction were sometimes lost when the server recovered. This has been fixed.

Reference # and Release	Description
3.1.0	Slow delivery of messages to a subscriber was occurring when <code>tibrvcn_sync_ledger</code> was set to <code>true</code> , <code>tibrv_topic_import_dm</code> was set to <code>PERSISTENT</code> , the subscriber was a durable subscriber, and the sender published a large number of messages very quickly. Until the server absorbed the messages, the server would not deliver the messages to the subscriber. This has been fixed.
3.1.0	The administration API was not able to clear all tracing flags. This has been fixed.
3.1.0	When the <code>set server ft_ssl_password=<string></code> command was issued (either using the administration tool or by way of the equivalent administration API calls), the server would erroneously report that the server rate collection interval was also changed. This has been fixed.
3.1.0	The administration API was not receiving SSL parameters for routes when calling <code>getRoutes()</code> . This has been fixed.
3.1.0	Under certain situations, the server crashed when compressed messages were exported to TIBCO Rendezvous. This has been fixed.
3.1.0	Under certain situations, transaction IDs were reset and transactions could be assigned a non-unique ID. This has been fixed.
Issues Closed in Release 3.0.1	
3.0.1	A Java client could deadlock if its connection to the server is terminated during execution of create session request. This has been fixed.
3.0.1	The server could leak memory in certain situations when processing messages imported from TIBCO Rendezvous. This has been fixed.
3.0.1	Now a Java client attempting to log in with a username and password can login when using SSL connections.
3.0.1	When sending large messages to the TIBCO Enterprise Message Service server, sometimes performance degraded. This has been fixed.
3.0.1	When several XA clients are committing and/or rolling back transactions at the same time, sometimes a deadlock in the <code>tibjmsd</code> occurred. This has been fixed.
3.0.1	The TIBCO Enterprise Message Service server did not rollback an active XA transaction if the client that started the XA transaction broke its connection to the server. This has been fixed.

Reference # and Release	Description
3.0.1	Messages imported from TIBCO Rendezvous were not bridged correctly. This has been fixed.
3.0.1	When RVCN messages were imported and <code>tibjmsd</code> was unable to store the JMS version of the message, data was sometimes lost. This has been fixed.
3.0.1	After a topic or queue was destroyed and then recreated, RVCN messages were not imported to that destination. This has been fixed.
3.0.1	RVCN messages imported to a JMS topic were not delivered to a topic subscriber in a timely manner under certain situations. This occurred when the RVCN ledger had pre-registered the server as a listener, but the topic listener had not been started. The messages were queued in the RVCN sender until the topic subscriber was started. However, the messages were only delivered to the subscriber one at a time as new RVCN messages were sent. This has been fixed.
3.0.1	If the flag passed to an XAResource's <code>recover</code> method in a Java client was anything other than <code>TMSTARTSCAN</code> , <code>recover</code> returned a null instead of an array of 0 length. This has been fixed.
3.0.1	When a subscriber exited a global topic, any queues that were bridged from that global topic could experience message loss. This has been fixed.
3.0.1	Sometimes empty queues had a non-zero Pending Message Size statistic. This has been fixed.
3.0.1	When the total number of producers created by clients exceeds 64,000, there could possibly be message corruption. This has been fixed.
3.0.1	Java Keystore files with extension <code>.JKS</code> and Entrust Store files with extension <code>.EPF</code> could not be specified in <code>ConnectionFactory</code> s. This has been fixed.
3.0.1	The <code>getJMSVersion</code> and <code>getJMSMinorVersion</code> methods on the <code>ConnectionMetaData</code> interface returned 1.0 and 0 instead of 1.1 and 1, respectively. This has been fixed.
3.0.1	Correct promotion of bridge and RV export interests for routed topics was prevented. This has been fixed.
3.0.1	Queue messages that were received but not acknowledged in a failed client were not redelivered immediately to other queue receivers. This has been fixed.

Reference # and Release	Description
3.0.1	The server may fail when an incorrect create destination request is sent by way of the Java Administration API. This has been fixed.
3.0.1	Delivery to queue receivers is now more fair. Messages are now delivered to each receiver in turn, provided they have not reached their prefetch limit. See Changes in Functionality on page 13 for more information.
3.0.1	C Client: In some situations, the client could incorrectly stop receiving messages while messages are available in the server. This has been fixed.
3.0.1	C Client: <code>tibjmsObjectMsg_GetObjectBytes</code> did not work correctly when used on just received message. This has been fixed.
3.0.1	C Client: If a C client rolls back an XA transaction, there could possibly be re-delivery of fully acknowledged messages. This has been fixed.
3.0.1	<p>C Client: When a C client called <code>tibjmsQueueConnection_CreateQueueSession()</code> with one of the following TIBCO-specific extensions for acknowledge mode, a <code>TIBJMS_ILLEGAL_STATE</code> exception was returned:</p> <ul style="list-style-type: none"> • <code>TIBJMS_NO_ACKNOWLEDGE</code> • <code>TIBJMS_EXPLICIT_CLIENT_ACKNOWLEDGE</code> • <code>TIBJMS_EXPLICIT_CLIENT_DUPS_OK_ACKNOWLEDGE</code> <p>This has been fixed.</p>
3.0.1	C Client: There was a flow-control defect that caused C clients to receive more messages than the prefetch size (both topics and queues). This has been fixed.
3.0.1	C Client: Calling <code>tibjmsBytesMessage_ReadBytes</code> on an empty bytes message incorrectly returns <code>TIBJMS_MSG_EOF</code> instead of <code>TIBJMS_OK</code> and return length -1. This has been fixed.
3.0.1	C Client: C client library for Linux 2.4 did not work on RedHat 7.2 platform. This has been fixed.
Issues Closed in Release 3.0.0	
3.0.0	SSL connection between the client and the server can now be established through the firewall.
3.0.0	Java client no longer fail to read empty BytesMessage.

Reference # and Release	Description
3.0.0	C client now correctly returns EOF error when reading last field of <code>tibjmsBytesMsg</code> in some cases.
3.0.0	When <code>ConnectionFactory</code> objects stored in LDAP are obtained by way of JNDI, they now correctly preserve null client ID.
3.0.0	If the server loses the network connection, Java clients that do not send messages now detect connection failure.
3.0.0	A Java client formerly failed if it can not read System Properties when running inside the browser. This has been fixed so that the Java client continues with default values.
3.0.0	Acknowledging a message after it has expired incorrectly failed. Message expiration only prevents a message from being delivered after it has expired. If a message was received before it expired, it can now be confirmed any time later.
3.0.0	Destination names with spaces are not enclosed in double quotes when written into <code>topics.conf</code> and <code>queues.conf</code> and subsequently the server failed to read the configuration. This has been corrected.
3.0.0	Route configuration can now be created over an existing passive route.

Known Issues

This section lists issues that are open in the named releases and provides workarounds where known.

Identified in Release	Defect #	Description	Workaround
4.3	1-6VASSK	Java API constructor <code>TibrvJMSTransport</code> throws a <code>NullPointerException</code> when <code>serverURL</code> is null and <code>emulateReconnect</code> is true.	Supply a non-null value for <code>serverURL</code> .
4.2	1-5DU9WW	<code>InstallShield</code> problems prevent uninstalling EMS from Linux 24gl23 Itanium platform.	Uninstall using this command line (all on one line): <pre>java -cp TIBCO_EMS_HOME/_uninst/uninstall.jar run</pre>
4.1.0	1-22ZRNM	JSSE cannot read PKCS12 certificates generated by some versions of OpenSSL.	Import the certificate into a web browser; then export the certificate to a new file with extension <code>.p12</code> (not <code>.pfx</code>).
4.1.0	1-2ZP5N5	When a client disconnects from the server, the server should automatically delete any temporary queues that the client created. If the temporary queue contains messages, then the server does not automatically delete it.	Administrators can manually delete such abandoned temporary queues from the server.

