TIBCO SmartSockets™

Release Notes

Software Release 6.8.0 July 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 SMARTSOCKETS INSTALLATION 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 Adapter, RTclient, RTserver, RTworks, SmartSockets, and Talarian are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or 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 © 1991-2006 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Release Notes 1	1
New Features	2
Release 6.8	2
Release 6.7	4
Release 6.6.0	6
Release 6.5.1	7
Release 6.5.0	7
Release 6.4.2	Э
Release 6.4.1	0
Release 6.3	2
Release 6.2	2
Release 6.1	4
Release 6.0	5
Deprecated Features	9
Release 6.8	Э
Release 6.7	Э
Release 6.6.0	С
Release 6.4.1	C
Release 6.2	1
Release 5.1	1
Migration and Compatibility	2
Closed Issues	3
Known Issues	4

iv | Contents

Release Notes

This document includes release notes for TIBCO SmartSockets[™].

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 a username, you can request one. You must have a valid maintenance or support contract to use this site.



The TIBCO Product Support web site contains late breaking news, information about cumulative fix releases, and an FAQ sheet. We recommend that all customers check these resources regularly.

Topics

- New Features, page 2
- Deprecated Features, page 19
- Migration and Compatibility, page 22
- Closed Issues, page 23
- Known Issues, page 34

New Features

This section lists features added since the last release of this product.

Release 6.8

Reference # and Release	Feature	
New Features in Release 6.8		
1-6WYFL8 1-6WSSFR 1-6RMDXF	Added support for Solaris 10 on x86 platforms (32- and 64-bit) and 64-bit support on all x86 platforms (Linux, Windows, Solaris 10).	
6.8		
1-6ZLBE7 6.8	Added file as a possible value for ipc_gmd_type. This value will cause calls to TipcSrvConnOpen (and similar functions) to attempt to open the GMD files upon opening a connection, and fail if there is a problem. Additional logging messages have also been added.	
1-PSDEP 6.8	Added an additional field to the MON_CLIENT_NAMES_STATUS message returned by the TipcMonClientNamesSetWatch and TipcSrvMonClientNamesSetWatch functions that describes the reason for the client's disconnection. The new field is also displayed by RTmon (if it is empty or not present, then "no reason given" is displayed). Refer to TipcMonClientNamesSetWatch in TIBCO SmartSockets Application Programming Interface for a list of possible values.	
1-6YE8K9 6.8	The rtinit.sh and rtinit.csh scripts have been enhanced to correctly set LD_LIBRARY_PATH (Solaris, Linux) or SHLIB_PATH (HP-UX) to include the directories for the 64-bit SmartSockets libraries so that this no longer needs to be done by hand. If you are not using these scripts, you should review them to ensure that your own configuration adds the correct paths. See the <i>TIBCO SmartSockets Utilities</i> guide for details.	
1-6TBSBA 6.8	Added disable_mon_watch_types option to RTserver. This option is a list of watches (as accepted by the 'watch' RTmon command) to ignore. See the <i>TIBCO SmartSockets User's Guide</i> for details.	

Reference # and Release	Feature		
New Features in F	New Features in Release 6.8		
1-5CRLI1 6.8	The connect, disconnect, subscribe and unsubscribe commands now support multiple connections in RTclients when used via control messages. The action of these commands will apply to the conn on which the control message was received.		
1-6TT43J 6.8	Added low-memory fault tolerance to RTserver. Certain operations likely to fail (especially when dealing with an unusually large message) now fail gracefully if their memory requirements cannot be fulfilled, by dropping either messages or inbound connections. "Important" (GMD, Persistent, or protocol) messages will still cause RTserver to abort if sending them fails (receipt of such a message will simply disconnect the conn the message was received on).		
1-6PZ6SV 6.8	Improved memory usage behavior of RTserver. This change allows a performance improvement that may be significant on some platforms.		
6.8	Changed allocation functions to print the size of the failed allocation in its call to TutFatalError (when a SmartSockets application runs out of memory). Note that this information will not be in the trace_file, and may only be available on the terminal.		
1-61RER1 6.8	Updates to the PGM link driver logging to improve diagnostics.		

Release 6.7

Reference # and Release	Feature		
New Features in F	New Features in Release 6.7		
1-I9E8W 1-11MXWC	New monitoring calls allow clients to poll for client version, server version, and server start time.		
6.7			
1-5165V1 6.7	Traffic statistics are upgraded to use 64-bit integers. New API entry points and polls reflect this feature.		
1-1AP7D3	ss.socket_connect_timeout is now available in Java (JRE 1.4 or later).		
6.7			
1-19MTRE	The Java API now supports timestamps as a trace_flags option.		
6.7			
1-37GUVN	fullzone is a new value for the time_format option (all language APIs). It		
6.7	specifies a full date and time string, including the time zone designation.		
1-1TQ0W4	_next is a new special value for the Server_Names option in RTclient. It		
6.7	specifies that after server failover, the reconnection algorithm continues scanning the list of server names at the next server in the list (instead of start at the beginning of the list each time).		
1-1YH5WB 6.7	New API calls get the logical connection name of the currently connected RTserver:		
	TipcSrvGetCurrentConnectedLcn		
	TipcSrvConnGetCurrentConnectedLcn		

Reference # and Release	Feature		
New Features in F	New Features in Release 6.7		
1-1JAM1D 6.7	New RTserver options implement enhanced multi-thread mode. See Multi-Thread Mode in the Options Reference chapter of <i>TIBCO SmartSockets User's Guide</i> .		
1-1PNCWX 6.7	Two new server options guide the server as it cleans up subscriptions from a destroyed client connection. When clients use a very large number of subscriptions, these options can increase server responsiveness.		
	Client_Drain_Timeout		
	Client_Drain_Subjects		
1-IBGLZ	.NET and Java now fully support the following options:		
6.7	• ss.trace_file		
	• ss.trace_file_size		
	• ss.trace_level		
	• ss.time_format		
	• ss.trace_flags		
	• ss.backup_name		

Release 6.6.0

TIBCO SmartSockets (plug-in) Palette for BusinessWorks 5.1.

You can now implement BusinessWorks[™] 5.1 processes using SmartSockets activities and resources. This release contains four (4) files pertinent to the SmartSockets palette for BusinessWorks:

- A JAR file \$RTHOME/java/lib/ss-bwp.jar containing the palette.
- A JAR file \$RTHOME/java/lib/ss-flavor.jar which patches BusinessWorks 5.1.13, so it correctly supports the SmartSockets palette.
- The SmartSockets Java API \$RTHOME/java/lib/ss.jar.
- The SmartSockets Palette for BusinessWorks README which contains installation and setup instructions \$RTHOME/doc/bwp/ssbwplugin.htm.

Fully Managed .NET Application Interface

TIBCO SmartSockets now includes a fully managed .NET API. The new .NET API is documented in the *TIBCO SmartSockets User's Guide and Tutorial*.

Increased RTserver Scalability

This release of SmartSockets includes two updates that will increase the scalability of a single RTserver.

- **64 Bit RTserver processes** You can now run RTserver processes as 64 bit processes, thus increasing the capacity of a single server to handle message load and RTclient connections. This support is available on the following platforms: Sun Sparc: Solaris 2.7 and up, Intel IA64 Linux, Itanium HPUX (IA64_HPUX), and HP 9000 Series: HP-UX 11.x.
- Increased RTclient connection limit You can now connect more RTclients to the same RTserver[™] than in the previous releases. In previous releases the number of RTclients that could connect to a single RTserver was limited by the number of file descriptors that could be added to a select mask. We have now added support for using the poll facility which now allows us to use as many file descriptors as the process is allowed by the OS. This support is available on the following platforms: Sun Sparc: Solaris 2.7 and up, Intel x86 and IA64 Linux, and HP 9000 Series: HP-UX 11.x.

Enhanced Monitoring

This release of TIBCO SmartSockets includes a new option, monitor_level, which allows for the enabling of none, some, or all of the monitoring calls. By default monitor_level is set to some, which enables all monitoring except traffic data monitoring. For example, set the monitor_level option as indicated below to enable traffic monitoring:

setopt monitor_level all

Improved RTserver Log File Information

RTserver will now print to the log file the IP address of connecting RTclient processes.

Release 6.5.1

• New Platforms Supported This release of TIBCO SmartSockets is available on an expanded platform list, including OpenVMS and Intel IA64 (Itanium). See the readme.txt file for more information.

Release 6.5.0

Authentication and Authorization

TIBCO SmartSockets now includes Basic Security. Basic Security provides a standard way to authenticate RTclients, using usernames and passwords. It also enables access control lists (ACLs) with the following authorizations:

- Server connections which RT processes, or users, can connect to RTserver
- Subject subscriptions which users can subscribe to a subject
- Subject publications which users can publish to a subject

The security is based on a username-password scheme and ACLs are stored in a file local to the machine that RTserver is running on.

Compression

TIBCO SmartSockets now offers message compression. Compression can greatly reduce the bandwidth needed to send messages over a network, but can cost additional CPU time. It is useful in situations where you need to transfer messages across lower bandwidth connections such as WANs, or when message data is large or text-based. There are two different compression configurations:

- Message level compress by message type, or on a per message basis. Only the message payload is compressed. The message is automatically decompressed when the receiver attempts to access the message data.
- Connection level causes all messages sent over a connection to be compressed. Messages are compressed before transmission over a connection, and decompressed upon arrival.

Monitoring Optimizations

Two new monitoring features were added to this release of SmartSockets:

• In previous versions of SmartSockets, all monitoring requests were routed through RTserver, even if the RTclient requests information on itself. RTclients are now able to monitor their own buffers without going through RTserver, removing unnecessary use of the network. The new functions supporting this feature are:

TipcConnTrafficGetBytesRecv	TipcConnTrafficGetBytesSent
TipcConnTrafficGetMsgsRecv	TipcConnTrafficGetMsgsSent
TipcConnBufferGetReadSize	TipcConnBufferGetWriteSize
TipcSrvConnTrafficGetBytesRecv	TipcSrvConnTrafficGetBytesSent
TipcSrvConnTrafficGetMsgsRecv	TipcSrvConnTrafficGetMsgsSent
TipcSrvConnBufferGetReadSize	TipcSrvConnBufferGetWriteSize
TipcSrvTrafficGetBytesRecv	TipcSrvTrafficGetBytesSent
TipcSrvTrafficGetMsgsRecv	TipcSrvTrafficGetMsgsSent
TipcSrvBufferGetReadSize	TipcSrvBufferGetWriteSize

• Extension data from RTclients, which is data created within an RTclient, can be monitored by another RTclient with the TipcMonExt* or TipcSrvMonExt* APIs. Poll with TipcMonClientExtPoll. The RTserver is not involved in generating this kind of monitoring information.

Server_Async_Subscribe Option

The Server_Async_Subscribe option specifies whether or not an RTclient waits for a response from RTserver after sending a new subscription request. RTclients connecting securely to RTserver, usually with usernames and passwords set with the TipcSrvSetCredentials or TipcSrvSetUsernamePassword function, can use this option to control whether the RTclient waits for confirmation from the RTserver.

RTclients that are not using TipcSrvSetCredentials or TipcSrvSetUsernamePassword are not affected by the Server_Async_Subscribe option.

See the *TIBCO SmartSockets User's Guide* for more information.

Support Removed

TIBCO SmartSockets no longer supports:

• rtlinkmon shell script no longer supported

The rtlinkmon shell script is no longer supported. You cannot link custom RTmon executables to optional libraries.

• rtmwm shell script no longer supported

Release 6.4.2

The following are new features in Release 6.4.2.

- TIBCO SmartSockets now supports the development of SmartSockets applications using the Microsoft Visual C++ .NET compiler.
- The initial shipment of SmartSockets now includes support for Tru64 and AIX in addition to Sun Solaris, Windows, Linux, and HP-UX.

Release 6.4.1

The following are new features in Release 6.4.1.

- TIBCO SmartSockets now includes 64-bit versions of the RTclient libraries as part of the initial release. 64-bit library support is available for HP-UX Version 11.0 and Solaris platforms.
- TIBCO SmartSockets now includes API function calls and message types that support advisory and monitoring messages. You can poll for the number of RTclients in a project, the number of subjects an RTclient is subscribed to, or RTclient or RTserver CPU usage. You can watch for RTclient or RTserver write buffer congestion, or for the RTserver cloud to reach and attempt to surpass the maximum licensed number of client connections. The new functions are:

Function for Global Connections	Function for Multiple Connections
TipcMonClientCongestionGetWatch	TipcSrvMonClientCongestionGetWatch
TipcMonClientCongestionSetWatch	TipcSrvMonClientCongestionSetWatch
TipcMonClientCpuPoll	TipcSrvMonClientCpuPoll
TipcMonClientInfoPoll	TipcSrvMonClientInfoPoll
TipcMonClientNamesNumPoll	TipcSrvMonClientNamesNumPoll
TipcMonClientSubscribeNumPoll	TipcSrvMonClientSubscribeNumPoll
TipcMonServerCongestionGetWatch	TipcSrvMonServerCongestionGetWatch
TipcMonServerCongestionSetWatch	TipcSrvMonServerCongestionSetWatch
TipcMonServerCpuPoll	TipcSrvMonServerCpuPoll
TipcMonServerMaxClientLicensesGetWatch	TipcSrvMonServerMaxClientLicensesGetWatch
TipcMonServerMaxClientLicensesSetWatch	TipcSrvMonServerMaxClientLicensesSetWatch

For detailed descriptions of these functions, see the *TIBCO SmartSockets User's Guide and Tutorial*.

- The *TIBCO SmartSockets User's Guide* contains a new section describing the SmartSockets dispatcher and event features. Dispatchers and events are used primarily with multiple connections.
- The new option Trace_File_Size sets the maximum size of a trace file, enabling circular trace files. This prevents a long-lived process from filling up the disk of the machine on which it is running.
- There are new RTclient options, Ipc_Gmd_Auto_Ack and Ipc_Gmd_Auto_Ack_Policy, that let you control the automatic acknowledgement of guaranteed message delivery (GMD) messages. For more information, see the *TIBCO SmartSockets User's Guide*.
- There is a new option, Monitor_Ident, that sets the monitoring identification string for a process. Identification strings can also be set with TipcMonSetIdentStr. For more information, see the *TIBCO SmartSockets User's Guide*.

The related SmartSockets Java client option is ss.monitor_ident. For more information, see the *TIBCO SmartSockets Java Library User's Guide and Tutorial*.

- This release supports the TIBCO SmartSockets Rendezvous Adapter, Software Release 1.0.0.
- This release of SmartSockets supports SmartSockets SSL, Release 2.1.1.
- TipcMsg.destroy is not supported by TIBCO SmartSockets. When acknowledging GMD messages, a Java client should call TipcMsg.ack instead of TipcMsg.destroy.
- The SmartSockets documentation library now includes the *TIBCO SmartSockets C++ User's Guide*, which describes the sscpp library for C++ users.

Release 6.3

The following are new features in Release 6.3.

- Bandwidth rate control for RTservers and RTgms processes lets you regulate RTserver and RTgms bandwidth usage for each connection. New options were introduced that you can set in startup command files, or dynamically using ADMIN_SET messages. For more information, see the *TIBCO SmartSockets User's Guide*.
- You can now set certain SmartSockets Multicast options dynamically using an ADMIN_SET message. This lets you set those options for a particular multicast group connection. For more information, see the *TIBCO SmartSockets User's Guide*.

Release 6.2

The following are new features in Release 6.2.

- This release is the first TIBCO SmartSockets release to support these new products:
 - TIBCO SmartSockets Monitor
 - TIBCO SmartSockets Cache
 - TIBCO SmartSockets LiveWeb
- SmartSockets Version 6.2 supports SmartSockets SSL Version 2.0 and higher. If you use SmartSockets SSL, you must upgrade from Version 1.
- The standalone SmartSockets product no longer includes HTTP CONNECT or SOCKS support. To connect through firewalls and proxy servers, you must purchase the SmartSockets LiveWeb product and use that in conjunction with SmartSockets. For more information, see the *TIBCO SmartSockets LiveWeb User's Guide*.
- SmartSockets now supports COM, in addition to ActiveX. The new COM APIs are documented in the *TIBCO SmartSockets ActiveX and COM Programmer's Guide*.
- Multiple connections to RTservers are supported now in C and C++, as an alternative to the single global connection. Previously, multiple RTserver connections were only supported in Java. Many APIs have been added for this support, including TipcSrvConn, TipcSrvMon, TipcMtLog, TipcEvent, and TipcDispatcher. For information on using multiple connections, see the *TIBCO SmartSockets User's Guide* and the API documentation for C and C++.
- There is a new set of C++ libraries, named sscpp. The new libraries contain the multiple connection support and also include new callback

implementations, new datatyping, and new utility classes. The original C++ libraries are also supported and shipped in this release. You can choose which set of libraries you want to use. The new libraries are not source-compatible and require code changes if you want to use them.

- You can now do license management by updating a license file instead of branding the server. Licenses for RTserver, RTgms, and SmartSockets products such as SmartSockets LiveWeb can be cut and pasted into the license file found in:
 - UNIX: \$RTHOME/standard/talarian.lic
 - Windows: %RTHOME%\standard\talarian.lic

For more information, see the TIBCO SmartSockets Installation Guide.

There is also a new license utility, rtlic, to simplify looking up your license information. Information on this utility is in the *TIBCO SmartSockets Utilities* reference.

- There are new client options, Log_In_Msgs and Log_Out_Msgs, that enable you to log all incoming or all outgoing messages for a client. For more information, see the *TIBCO SmartSockets User's Guide*.
- These C APIs can be used to manipulate subjects: TipcSubjectGetUnique, TipcSubjectMatch, and TipcSubjectValid. For more information, see the *TIBCO SmartSockets User's Guide and Tutorial* reference.
- There is a new Java method, Tut.affirm, that replaces Tut.assert. This is the first step in deprecating Tut.assert in the Java class library. With JDK 1.4, Sun has introduced assert as a Java language keyword. To be compatible, the SmartSockets Java class libraries must stop using Tut.assert.

For this release, you can still use Tut.assert but you receive deprecation warnings when you compile. In a future release, Tut.assert will be removed completely.

We recommend you begin using Tut.affirm instead of Tut.assert.

Release 6.1

The following are new features in Release 6.1.

- This release supports the SmartSockets for JMS product, which replaces the Workbench for JMS product.
- When you start the RTserver using the rtserver shell script, the default is to start the optimized version of RTserver. You can use the -check argument of rtserver to start the non-optimized version of RTserver, which has additional validations and checking. For more information, see the *TIBCO SmartSockets User's Guide*.

A new argument has been added to the rtserver shell script called -no_console. Specifying this argument prevents the start of an associated Windows console when you start RTserver as a detached process. For more information, see the *TIBCO SmartSockets Utilities* reference.

- Three new RTserver options have been added that enable you to limit the number of server-to-server connections for an RTserver. For example, you could configure an RTserver that can request connections to other RTservers but does not allow connections initiated by other RTservers. The options, Max_Server_Conns, Max_Server_Accept_Conns, and Max_Server_Connect_Conns, are described in the *TIBCO SmartSockets User's Guide*.
- The extension to logical connection names that enables you to connect through a firewall or proxy server with HTTP CONNECT enabled has been modified. Instead of specifying http, you must specify http_connect, as in this logical connection name:

http_connect:www.company.com:80@tcp:1.2.3.4:5101

For more information, see the section on tunneling in the *TIBCO SmartSockets* User's Guide.

• A new delivery mode, T_IPC_DELIVERY_ORDERED, is available for messages sent through SmartSockets Version 6.1 RTservers. It is not a guaranteed message delivery (GMD) mode, but does perform better during a network or process failure than the delivery mode T_IPC_DELIVERY_BEST_EFFORT. Like BEST_EFFORT, messages can still be lost during a network or process failure. Unlike BEST_EFFORT, ORDERED ensures that even during a failure, messages arrive in the order in which they were published.

The ORDERED delivery mode was added to support JMS clients, because JMS clients cannot handle receiving messages out of order. If you have C clients publishing to JMS receiving clients, set the message delivery mode to ORDERED instead of to BEST_EFFORT.

To guarantee delivery, GMD makes a copy of each published message that can be re-sent in event of a failure. This requires more resources than non-GMD messaging, and slows performance. Consider using the ORDERED delivery mode as an alternative to the GMD delivery modes of SOME or ALL for applications where order, even in the event of a failure, is critical but the application can tolerate possible message loss during failures in exchange for better performance than you can achieve with GMD.

All RTservers and RTclients handling messages with a delivery mode of ORDERED must be at a Version 6.1 level or higher.

For more information on this message property, see the *TIBCO SmartSockets User's Guide*.

- By default, the SmartSockets for JMS standard message types:
 - JMS_BYTES
 - JMS_MAP
 - JMS_OBJECT
 - JMS_STREAM
 - JMS_TEXT

now have a delivery mode of T_IPC_DELIVERY_ORDERED.

• The SmartSockets documentation library is no longer included with the product bundle. The online documentation files in HTML and PDF formats are available in a separate bundle on the FTP site.

Release 6.0

The following are new features in Release 6.0.

• Firewall tunneling

SmartSockets supports tunneling through a firewall, allowing you to specify a proxy server as part of your logical connection name, and providing RTclient options that specify a proxy user ID and password if one is needed for authentication. For more information, see the *TIBCO SmartSockets User's Guide*.

• Updating and deleting named fields

SmartSockets supports updating and deleting named fields, expanding the functions you can perform on named fields. For more information, see the *TIBCO SmartSockets User's Guide and Tutorial* reference.

• New message fields

There are new message fields, including fields for message IDs and correlation IDs, and new APIs for getting and setting the values for the fields. For more information, see the *TIBCO SmartSockets User's Guide and Tutorial* reference.

• New options with the rtserver command

The rtserver command was streamlined, and the arguments you can specify when you issue the command to start or stop the RTserver have changed. For more information, see the *TIBCO SmartSockets User's Guide*.

• Improvements to tracing and problem diagnosis

All trace and debug information is stored in a single file. You can specify the name of the file as an argument on the rtserver command, or you can use the Trace_File option. You can specify what level of trace information you want logged to the trace file, either as an argument on the rtserver command or with the Trace_Level option. You can use the Trace_Flags option to set the prefix for the format of the trace information. For more information, see the *TIBCO SmartSockets User's Guide*.

• Specifying file-based or memory-based Guaranteed Message Delivery (GMD)

There is a new RTclient option that allows you to specify file-based or memory-based GMD. For the C API, see the information in the *TIBCO SmartSockets User's Guide* on the Ipc_Gmd_Type option. For the Java API, see the information in the *TIBCO SmartSockets Java Library User's Guide and Tutorial* on the ss.ipc_gmd_type option.

Improved SmartSockets documentation

The SmartSockets documentation library was improved in this release. The look of the books was redesigned for better readability, and you can view the books online in either PDF or HTML format. The PDF and HTML files are included with the product on the CDROM and FTP distribution set.

• Multiple RTserver connections for Java

In previous releases, an RTclient could only establish a connection to one RTserver at a time. In this release, in Java, an RTclient can connect to multiple RTservers. Multiple RTserver connections are useful when:

- threads such as Java applets or servlets might require individual connections to register independent subscriptions and callbacks
- threads such as Java applets or servlets might require individual connections for proper remote procedure call (RPC) handling

For more information, see the *TIBCO SmartSockets Java Library User's Guide and Tutorial*.

• Warm connections in Java

In previous SmartSockets releases, an RTclient could establish a warm connection to an RTserver using the C API. In this release, warm connections are also supported by the Java API. For more information, see the *TIBCO SmartSockets Java Library User's Guide and Tutorial*.

• New RTserver options

Two new RTserver options have been added, Client_Read_Timeout and Client_Keep_Alive_Timeout. The purpose of these options is to allow keep alive traffic to be maintained between the RTserver and RTclient. For more information, see the *TIBCO SmartSockets User's Guide*.

• RTserver command mode

RTserver can no longer be started in interactive command mode. The -cmd_mode argument is no longer supported.

• RTservers can listen on all IP addresses on a machine

Previously, if an RTserver was on a machine with multiple IP addresses, the RTserver listened only on the default IP address. Now you can use the new keyword _any to cause the RTserver to listen on all IP addresses on a machine. For more information, see the *TIBCO SmartSockets User's Guide*.

• New callback and new API functions added

A new callback, the Server_Names Traverse Callback, has been added for RTclients. This callback executes before a connection to each name specified in Server_Names is attempted. To support this callback, the functions TipcSrvTraverseCbCreate and TipcSrvTraverseCbLookup have also been added. For more information on using the callback, see the *TIBCO SmartSockets User's Guide*. For information on using the API functions, see the *TIBCO SmartSockets User's Guide and Tutorial* reference.

• New utility function

You can use a new utility, TutThreadDetach, to detach a thread. For more information, see the *TIBCO SmartSockets Utilities* reference.

SmartSockets compatibility

This release of SmartSockets does not support RTworks[™] Version 4 or Version 3.5.

Changes were made to the RTserver to enhance GMD performance. Because the way an RTserver supports GMD messages has changed in Version 6.0, all the RTservers involved in GMD messaging should be SmartSockets Version 6.0 Revision 2 or higher. For non-GMD transactions, SmartSockets Version 6.0 RTservers are compatible with Version 5.5 RTservers. Releases of RTservers earlier than Version 5.5 are not supported with Version 6.0 RTservers.

SmartSockets Version 6.0 RTservers are supported with Version 5.5 and 5.2 RTclients.

• SmartSockets Version 6.0 does not support HP-UX 10.20

This release of SmartSockets does not support HP-UX 10.20. To use SmartSockets Version 6.0 on an HP system, you must be running HP-UX 11.0 or higher.

• Changes to the rtserver command replace the rtlicense shell script

The rtlicense shell script is no longer supported. Instead, use the rtserver command with the -license argument. For more information, see the *TIBCO SmartSockets User's Guide and Tutorial* reference.

• rtlinkserver shell script no longer supported

The rtlinkserver shell script is no longer supported. You cannot link custom RTservers to optional libraries.

• Init_Connect_Timeout option no longer supported

The RTserver option, Init_Connect_Timeout, is no longer supported. If you specify this option in a command file or using setopt, the option is ignored.

• Udp_broadcast is no longer a default in the Conn_Names option

The Conn_Names option no longer includes udp_broadcast as one of the default protocols. If you require the udp_broadcast protocol for your RTservers, you must change the value for Conn_Names to include udp_broadcast.

Deprecated 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 6.8

Reference # and Release	Feature
Deprecated in Release 6.8	
1-72S0P2 1-70SI3Z 6.8	The TIBCO SmartSockets [™] COM API has been deprecated and will be removed in a future version. The <i>TIBCO SmartSockets ActiveX and COM Programmer's Guide</i> is available by special request.
6.8	The GDI portion of RTmon (rtmongdi) has been deprecated and will be removed in a future version.
	Deprecated the 'stats' logging option in the PGM link driver.

Release 6.7

Reference # and Release	Feature
Deprecated in Rel	ease 6.7
1-5I65V1 6 7	Traffic statistics are upgraded to use 64-bit integers. New API entry points and polls reflect this feature.
0.1	New forms (with the suffix 8) supersede these older forms:
	• TipcSrvTrafficGet*
	• TipcConnTrafficGet*
	• TipcSrvConnTrafficGet*
	Older forms remain for source backward compatibility only.

Reference # and Release	Feature
Deprecated in Rel	ease 6.7
1-1JAM1D 6.7	Enhanced multi-thread mode supersedes the deprecated option Server_Num_Threads.
	See Multi-Thread Mode in the Options Reference chapter of <i>TIBCO SmartSockets User's Guide</i> .

Release 6.6.0

The following features are deprecated in this release>

- ActiveX API The ActiveX API should no longer be used for RTclient development. Use the new .NET API in place of ActiveX.
- Platform Support The following platforms will no longer be supported with the next release of SmartSockets: IBM RS6000: AIX 4.3, Windows NT, Sun Sparc: Solaris 2.6.

Release 6.4.1

The following features are deprecated in this release.

TipcMsg.destroy

When acknowledging GMD messages, a Java client should call TipcMsg.ack instead of TipcMsg.destroy. For this release, you can still use TipcMsg.destroy, but you receive deprecation warnings when you compile. In a future release, TipcMsg.destroy will be removed completely. We recommend you begin using TipcMsg.ack instead of TipcMsg.destroy.

TipcSrvConnTraverseCbCreate

TipcSrvConnTraverseCbCreate is identical to the function TipcSrvTraverseCbCreate. For this release, you can still use TipcSrvConnTraverseCbCreate, but it will be removed in a future release. We recommend you use TipcSrvTraverseCbCreate instead.

TipcSrvConnTraverseCbLookup

TipcSrvConnTraverseCbLookup is identical to the function TipcSrvTraverseCbLookup. For this release, you can still use TipcSrvConnTraverseCbLookup, but it will be removed in a future release. We recommend you use TipcSrvTraverseCbLookup instead.

Release 6.2

The following features are deprecated in this release.

Tut.assert

In SmartSockets Version 6.2, there is a new Java method, Tut.affirm, that replaces Tut.assert. This is the first step in deprecating Tut.assert in the Java class library. For this release, you can still use Tut.assert but you receive deprecation warnings when you compile. In a future release, Tut.assert will be removed completely. We recommend you begin using Tut.affirm instead of Tut.assert.

Release 5.1

The following features are deprecated in this release.

TipcPacketToMsg and TipcMsgToPacket

The functions TipcPacketToMsg and TipcMsgToPacket are not supported after SmartSockets Version 5.1. Use TipcBuf* functions and TutBuf* functions instead. For an example of using TipcBuf*, see the section on advanced use of message files in the *TIBCO SmartSockets User's Guide*.

- TipcMsgGetLocalDelivery
- TipcMsgSetLocalDelivery
- TipcMtGetLocalDelivery
- TipcMtSetLocalDelivery

Migration and Compatibility

This section explains how to migrate from a previous release to this release.

If you currently have only SmartSockets installed, and are upgrading to the newest version, you can simply follow the instructions in the *TIBCO SmartSockets Installation Guide*. On UNIX, you must install into a new directory, with the default name being ss68. On Windows, you must un-install your current SmartSockets product before you install this new version. By default, the new version is installed into a program folder called ss68.

If you have other SmartSockets products also installed, such as SmartMQ or SmartSockets RTgateway, and you want to use them with this new SmartSockets Software Release 6.8 installation, you must re-install them into the new ss68 directory after you have completed your SmartSockets Software Release 6.8 install. First, check that the version and revision of your existing products are compatible with SmartSockets Software Release 6.8. Use the Compatibility Matrix provided with your online documentation or contact TIBCO Product Support to find out. In some cases, you might need to upgrade your other SmartSockets products.

For more information on upgrading your TIBCO SmartSockets installation, see the *TIBCO SmartSockets Installation Guide*. There are special upgrading instructions if you are:

- using TIBCO SmartSockets RTgateway to build MSLs (Message Source Links)
- upgrading a C++ application using the sscpp library
- upgrading from a TIBCO SmartSockets Version 5.x release
- using GMD with any RTservers that are TIBCO SmartSockets Version 6.0 Revision 1 or lower

Closed Issues

Reference # and Release	Summary		
Issues Closed	Issues Closed in Release 6.8		
1-76DHSM	Fixed a problem where a message with very large header extensions (possibly due to a large number of named fields) could crash the conn API. This affected both		
6.8	RTserver and RTclients.		
1-72IKME	Fixed a potential deadlock with multi-threaded Java and C# clients where one client is sending GMD/SOD messages while another one is receiving messages		
6.8	and the client is being watched for msg_recv or msg_send.		
1-6RONKG	Fixed a problem in the RTserver that could cause lost messages when publishing		
6.8	suspended indefinitely.		
1-744Z7J	Changed the memory model used on HP-UX to allow RTserver to use more than 0.9		
6.8	gb of memory.		
1-73744B	Fixed a timing issue in Java that could cause any method attempting to read a message (mainLoop next, search, etc.) to ignore its timeout parameter and		
6.8	become suspended until data arrives.		
1-717E11	Fixed a problem with TipcSrvConnMsgNext, TipcConnMsgNext and		
6.8	indefinitely if a steady stream consisting of only control messages (such as GMD_ACK's) is received.		
1-70564T	Fixed a possible race condition in RTgms that could lead to data transmission being		
6.8	indefinitely suspended.		
1-6Z0D89	Fixed a problem that could cause clients to exit under rare circumstances after losing their connection		
6.8			
1-6XGOIA	Fixed a bug in the compression link driver that could cause the conn to enter an inconsistent state and exit RTserver if large messages are published to a compressed		
6.8	link.		

The table in this section list issues that were closed in the named releases.

24 | Release Notes

Reference # and Release	Summary
1-6WSTIL 6.8	Fixed a bug that could cause subscribeEx to lose its message to a mainLoop operating on the same conn in a different thread. This problem was primarily evident in Java and C#.
1-6W5T0T 6.8	Limited the size of conn write_buffer nodes that will be pooled to reduce memory overhead in an environment where large messages are being sent.
1-6TQCF3 6.8	Fixed a problem with compression that caused latency on the data stream (data already received might not be uncompressed until additional data was received). In some cases, this could cause handshakes to fail, leading to clients that would time out when trying to connect.
1-6RTT0Z 6.8	Fixed a problem that caused RTserver to core on startup if -threads 1 was used with a license that does not allow multi-threaded servers.
1-6UWZ00 6.8	Fixed a problem in RTserver that prevented clients connected via RTgms from being disconnected correctly if the connection to RTgms was lost. This only affected RTserver in single-threaded mode.
1-6T3TUX 6.8	Fixed a problem with client_msg_send/client_msg_recv watches with Java and C# clients that required ss.monitor_level to be set to 'all' on the watched client for these watches to work.
1-6MBQI2 6.8	Fixed a problem with subjects ending in '/' that could cause RTserver to exit.
1-1B3HSF 6.8	Fixed a problem in the RTserver that could cause it to exit under particular circumstances involving mixed GMD and non-GMD messages and RTservers disconnecting from each other.
1-6PMEEZ 6.8	Fixed a timing problem that could cause RTclients connecting using the PGM link driver to core. The problem existed on all platforms, but was more prominent on Linux.
1-6MBQH9 6.8	Fixed a problem in the Java API that could cause deadlocks when a subscriber disconnects while publishing GMD messages.
Issues Closed	in Release 6.7
1-1AP7D3 6.7	Enhanced Java socket connect timeout to automatically work as it does in C, when JRE 1.4 or higher is present.

Reference # and Release	Summary
1-5UIW3K	Fixed a potential memory leak in RTserver when reconnecting warm clients.
6.7	
1-17BCE9	Enhanced documentation for RTserver <pre>server_num_threads</pre> option.
6.7	
1-5I65V1	Upgraded traffic statistics to use 64-bit integers.
6.7	
1-5ECC8Q	Resolved conflict with ATL libs on Visual Studio 7.
6.7	
1-54KS45	Fixed a C# API defect involving _random in server_names. This defect was
6.7	introduced in CFIX 5.
1-1TSYCG	Fixed a defect in which message traffic statistics in the server could overflow and
6.7	return a negative value.
1-5ECC8A	Fix a defect in the PGM link driver to prevent socket handling errors that could
6.7	potentially cause other errors.
1-540448	Update to prevent synchronization issues and memory leaks when destroying
6.7	events using a detached dispatcher.
1-5ECC82	Update to destroy subjects that expire with gmd_publish_timeout, whether or not a
6.7	local publisher is still attached.
1-5ECC7M	Update to trace message regarding thread usage to avoid error message on HPUX
6.7	versions prior to 11.11.
1-4RTA62	Fixed a memory leak in the dispatcher destroy code.
6.7	
1-4TR76X	Fixed a bug in subject matching logic when wildcard * is used for publishing or subscribing. This defect was introduced in introduced in CFIX 5.
6.7	

Reference # and Release	Summary
1-4TRGU9 6.7	Fixed a defect in which adding an unopened connection to a dispatcher caused an internal exception while trying to remove the open connection during dispatcher destroy.
1-4UPRHS 6.7	Fixed a defect in which an rtclient, disconnected from RTgms due to keepalive failure, might be unable to reconnect.
1-4MIIS1 6.7	Optimized internal subject tables to increase efficiency of unsubscribing subjects covered by a wildcard subscribe.
1-4MIIST 6.7	Fixed a defect in which GMD initialization sequence might stall clients indefinitely.
1-4MIIT7 6.7	Fixed a defect associated with server sequence number rollover.
1-4MIIT7 1-26JPWD 6.7	Fixed a defect associated with using <code>connect_one_stop</code> option in <code>lcn</code> list when the first server in the list is not available.
1-4MIITS 6.7	Update to GMD/Load Balancing to prevent a possible lost message in a multiple server cloud when the first message on a subject is published.
1-4AVDSL 6.7	Fixed a defect in the C# client library associated with server_names traversal.
SR-3B48P5 6.7	Fixed a defect in server subscribe routing, associated with wildcard subscribe on clients. This issue caused servers to not route messages to clients on remote servers when the wildcard subscriptions were removed.
SR-3PR8GH 6.7	Fixed a deadlock problem in the Java API associated with automatic acknowledgement of duplicate GMD messages and multiple threads using the same connection.
6.7	Fixed a memory leak that could occur in an RTclient when multiple threads access a connection to RTserver that is also attached to a dispatcher.
SR-35MQ41 6.7	Modified rtmon runtime so it can set the monitor_level option.

Reference # and Release	Summary
SR-32SHSX 6.7	Fixed a defect in SmartSockets Cache associated with logging.
SR-2YTJ2D 6.7	Removed a restrictions so that SmartSockets applications can run in the absence of a valid message catalog file.
6.7	Changed the locking behavior of named fields to improve performance.
SR-2ZZVQK	Modified the usage of server_read_timeout in Java.
6.7	Previously setting server_read_timeout affected the waiting time for a conn_init response. This parameter no longer affects that operation.
SR-2RH659	Improved Java memory management of the output stream buffer.
6.7	
SR-2G7FZ1	Fixed a defect in which some socket error codes could cause RTserver to stop
6.7	transporting data while consuming many CPU cycles.
SR-2IHYHX	Fixed a defect in the Java client library, in which it checked timeouts before
6.7	checking the read queue. As a result, it sent keepalives even though data was actually available on the queue.
SR-2IHYHX	The receiving queue size within the multicast client library is now user
6.7	comgurable.
SR-2L5N3H	Fixed a defect associated with reconnection—locking a connection just before a
6.7	sleep call. As a result, the connection was not serviceable until the sleep call returned.
SR-26DAL9	Fixed a defect affecting detached dispatchers, in which erroneous mutex behavior
6.7	caused deadlock.
SR-2MK52P	Fixed a defect in which certain monitoring watch calls could cause server crashes in a large server cloud.
6.7	
SR-2IDDBZ	Modified the default value of trace_file_size to 1 gigabyte. Previously, RTserver would allow the trace file to grow until a SIGXFSZ caused it to exit.
6.7	

28 Release Notes

Reference # and Release	Summary
1-26W3Q5	Fixed a defect in which message routing would cease when the sequence number
6.7	overflowed a four-byte integer $(2^{31}-1 = 2147483648)$.
SR-1TPM6P	Fixed a defect in which the T_INT4 monitoring traffic statistic counters could
6.7	overflow and return a negative value. Traffic counts now reset to prevent overflow.
1-23QYWT	Fixed a defect in which RTgms would receive a header, then loop waiting for the
6.7	data.
1-22GL5P	Fixed a defect in which the TIBCO.ss.dll did not recognize real8 values read
6.7	from the wire as legitimate numbers.
6.7	Removed a dual entry in the pgmopts.def file, which had caused an erroneous log message.
SR-21BZNW	Fixed a locking issue in the dispatcher which had caused a memory leak.
6.7	
1-24QQ4F	Fixed TutGetVersionNumber to return the correct version number.
6.7	
SR-23V75D	Fixed a COM library defect in which a delete on the dispatcher conn_array caused
6.7	the application to exit.
Issues Closed	in Release 6.6
1-ZIXPR	If available, RTserver log output now includes the IP address of the connecting
6.6.0	RTclients.
1-PYUC4	TIBCO SmartSockets now performs an automatic backup of server log files on
6.6.0	Windows systems. (This feature was previously only available for UNIX systems.)
SR-1XAZG1	Added new monitoring option, monitor_level, which allows you to exclude
6.6.0	traffic monitoring.
1-1WGIU5	Fixed problem with the caching of very large batch messages causing RTserver to exit. By default, messages that include more than 10 k of data will not be cached.
6.6.0	

Reference # and Release	Summary
1-1TSYEN 1-1TOZ34	New options added to the rtserver to indicate that monitor calls to watch subject names or client names will only return updates to the list and not the entire lists with each status message.
6.6.0	
1-KYHDE	Change to the SmartSockets dispatcher logic to service all connections equally ever
6.6.0	when one connection has a very high amount of traffic.
1-1TQZ1Y	Fixed a problem that resulted in the accumulation of redundant client data in
6.6.0	messages.
1-1PTK08	Fixed problem with subject unsubscribe process in the RTserver that can cause the
6.6.0	Riserver to exit when unsubscribing to whiceard subjects.
1-1L8LEL	Fixed problem with messages being routed to unsubscribed clients.
6.6.0	
1-1EU1NA	Fixed problem with authentication that denied access to users on local connections.
6.6.0	
1-18ZZGE	Changed trace level on End of File messages to info.
6.6.0	
Issues Closed in Release 6.5.1	
1-103BC3	Corrected a problem that caused a multi-threaded RTserver to receive an error
6.5.1	to server priming failed because of an unexpected interrupt.
1-10N0EX	Two or more RTgms processes using the same UDP encapsulation on the same
6.5.1	machine exhibited unpredictable behavior, because NAK messages were not always returned to the correct RTgms process. This has been fixed.
1-11YUF1	Fixed problem where GMD messages were lost when RTclients disconnected from
6.5.1	the Kiserver thoug and reconnected to a different Kiserver.
1-12KH3X	Fixed a problem that caused RTgms to fail when attempting to connect to the second RTserver specified in its Server_Names option.
6.5.1	

Reference # and Release	Summary
1-130E7F	In Java, messages sent with a delivery mode of T_IPC_DELIVERY_ORDERED can be lost when RTserver fails and restarts. The problem has been fixed.
6.5.1	
1-13K8DG	A Java RTclient can now make upd_broadcast requests to discover RTserver.
6.5.1	
1-155U58	RTserver could exit when the Server_Disconnect_Mode was set to warm in the
6.5.1	RImon command file. This has been fixed.
1-155U6O	Corrected TipcDispatcherMainLoop, which was not processing keep alive
6.5.1	messages.
1-16ZP4P	Synchronous subscriptions, where the Server_Async_Subscribe option is FALSE,
6.5.1	can fail with error type mismatch. This has been corrected.
1-18RECG	RTserver exit ed when multiple threads used rotating log files and high trace levels.
6.5.1	This has been fixed.
1-1B3HS2	RTgms failures can cause Java applications to hang when there are multiple
6.5.1	connections to more than one RTgms process. This has been fixed.
1-1B3HSF	Multi-threaded RTservers can exit due to a corrupt internal message property. This
6.5.1	has been fixed.
1-1F5T3D	rtadmin.exe and rtadmind.exe were linked to the debug version of the
6.5.1	SmartSockets, and could not be run unless it was installed. This has been fixed.
1-1HMSPD	RTclients changing subscription status for generic subjects, such as /, could see
6.5.1	temporary performance problems, especially with large subject namespaces. This has been fixed.
1-Z8XJR	Renumbered the monitoring message types
6.5.1	T_MT_MON_CLIENT_NAMES_NUM_POLL_CALL and T_MT_MON_CLIENT_NAMES_NUM_POLL_RESULT to follow normal SmartSockets monitoring message type numbering conventions. Code that used these message types in SmartSockets 6.5.0 must be recompiled to work correctly with SmartSockets 6.5.1 and above.

Reference # and Release	Summary
1-ZW5D3	Use of the encode callback can cause an assertion if the size of the message buffer is changed during the encoding process. The processing has been changed to calculate the buffer sizes after the encode callback has run.
6.5.1	
Issues Closed in Release 6.5	
1-IAFRH	TipcDispatcher was not returning error code when used incorrectly. This has been
(PDR 2348)	fixed.
6.5.0	
1-IAFRV	TipcSrvConnGmdFileDelete would core dump after a connection was established.
(PDR 2334)	This has been fixed.
6.5.0	
1-IBM6A	In Java, the destroy callback was not called when the RTserver went down. This has
6.5.0	been fixed.
1-IBM75	The COM method TipcDispatcherCreateDetached() did not correctly create a
6.5.0	detached dispatcher.
1-ICBE5	There were a number of SSCPP compilation warnings on Linux. This has been
(PDR 2295)	fixed.
6.5.0	
1-IFHP4	The sample benchmark program srvbench.c had possibility of an overflow on high
6.5.0	speed systems. This has been fixed.
1-IG114	A Java client's ss.unique_subject and ss.default_subject_prefix options should be read-only when there is a warm or full connection to the RTserver. A Java client's ss.project option should be read-only when there is a full connection to the RTserver. This was not the case, but has been corrected.
6.5.0	
1-IHV7Y	RTserver write buffer performance under heavy loads was improved.
6.5.0	
1-ILTTA	The Java client's ss.default_msg_priority was not taking affect.
6.5.0	

Reference # and Release	Summary
1-IR40P	In Java, the control callback was not recreated after reconnecting from a full
6.5.0	disconnect. The callback is now recreated.
1-IR418	In Java, the server_create and server_destroy callbacks were not recreated after a
6.5.0	full disconnect. They are now recreated.
1-IR41B	Java clients should remove subject subscriptions when fully disconnected from
6.5.0	RIserver; subject subscription were saved after a full disconnect from RIserver (server_disconnect_mode is warm).
1-J43OL	In Java, TipcSrv.isRunning and TipcSrvGetConnStatus did not return accurate
6.5.0	connection status. This has been fixed.
1-J43PC	A subscriber using GMD or load balancing, when failing over to a new server, did
6.5.0	not always receive messages. If a subscriber connected to a server that had no dir clients subscribed to the GMD or load balancing subject at the time the publishe began publishing, the subscriber lost messages.
	The subscriber now receives the messages.
1-K8MK4	RTgms does not set its ident variable. Fixed to set ident to RTgms.
6.5.0	
1-KIEVD	The peer-to-peer examples %RTHOME%\examples\sscpp\conn_clt.cxx and
6.5.0	conn_srv.cxx did not run correctly when built with VC 6.0 and VC 7.0, because stdout to a message file is not available in VC 7.0. The examples were changed to
	call the Tmsg class print method.
1-L0N5R	During a peer-to-peer connection (conn), TipcConnClientDestroy hangs. This was
6.5.0	fixed.
1-LSQV9	When the server disconnected a client due to a server-to-client keep alive failure,
6.5.0	the server did not put any message in the server log file, even when the trace level was set to verbose. This has been corrected.
1-LSQVO	When tracing is enabled on the client side, the log file did not contain the timing
6.5.0	information. Timing information is now available by default.
1-MB9AO	In RTmonitor, the -help command argument still displayed the obsolete -license as a valid RTmon commandhelp no longer displays the command.
6.5.0	

Reference # and Release	Summary
1-NDKOL	In both C/C++ and Java, if the unique subject is not set, GMD was file-based rather
6.5.0	than memory-based. This has now been corrected to use memory-based GMD under those circumstances.
1-NT2ED	In examples lesson 4, the java callbacks example svrcbs.java did not continue to attempt connecting after a connection to RTserver was lost. The example was corrected to repeat the connection attempt.
6.5.0	
1-PO44T	In the C++ class library, setDecryptObj, getDecryptObj, setEncryptObj, and
6.5.0	getDecryptObj did not work. The problem has been corrected.
1-QI3PQ	This misleading error message:
6.5.0	TAL-SS-15363-I Rejecting group <id =="" {xxx}="">. Server not enabled for multicast.</id>
	Has been updated to read:
	TAL-SS-15363-I Rejecting group <id =="" {xxx}="">. Server not enabled for group connections.</id>
1-TW5PG	Resolves stack overflow when using ITipcMsg::NextStrArrway().
6.5.0	
1-TZK58	A deadlock occurred in the Java client library and TCP protocol when a connection attempt was made in the error callback. This was fixed to prevent deadlock.
6.5.0	
1-TZK5L	Problems in setting the options for simultaneous connections in Java for PGM
6.5.0	caused deadlocks. The PGM link driver was updated to handle named options and multiple connections.
1-UY4YD	Fixed a problem causing RTgms to exit on Windows.
6.5.0	

Known Issues

The table in this section lists known issues in this release.

Defect #	Summary/Workaround
1-22GL5P	.NET Application Interface
	Summary There is a problem converting some 8 byte real numbers (doubles) from incoming messages. We are currently researching this issue and working with Microsoft for a resolution.
	Workaround None.
1-ICLHE	Project Names in Multicast
(was PDR 1588)	Summary It is possible for a multicast receiver to receive messages belonging to a different project. If there are multiple publishers with different project names sending messages to the same subject, a multicast client subscribing to that subject receives messages independent of the project.
	Workaround Assign separate group names to RTgms.