

Guide to Informix MaxConnect

Version 1.0
April 2000
Part No. 000-6678

Published by Informix® Press

Informix Corporation
4100 Bohannon Drive
Menlo Park, CA 94025-1032

© 2000 Informix Corporation. All rights reserved. The following are trademarks of Informix Corporation or its affiliates, one or more of which may be registered in the United States or other jurisdictions:

Answers OnLine™; C-ISAM®; Client SDK™; Cloudconnector™; Cloudsync™; Cloudview™; DataBlade®; Data Director™; Data Mine™; Data Mine Builder™; Decision FastStart™; Decision Frontier™; Decision Solution Suite™; Decisionscape™; Dynamic Scalable Architecture™; Dynamic Server™; Dynamic Server™, Developer Edition™; Dynamic Server™ with Advanced Decision Support Option™; Dynamic Server™ with Extended Parallel Option™; Dynamic Server™ with MetaCube®; Dynamic Server™ with Universal Data Option™; Dynamic Server™ with Web Integration Option™; Dynamic Server™, Workgroup Edition™; Dynamic Virtual Machine™; Enterprise Decision Server™; Formation™; Formation Architect™; Formation Flow Engine™; Frameworks for Business Intelligence™; Frameworks Technology™; Gold Mine Data Access®; i.Foundation™; i.Intelligence™; i.Reach™; i.Sell™; Illustra®, Informix®, Informix® 4GL; Informix® COM Adapter™; Informix® Extended Parallel Server™; Informix® Informed Decisions™; Informix® InquireSM; Informix® Internet Foundation.2000™; InformixLink®, InformiXML™; Informix® Red Brick® Decision Server™; Informix Session Proxy™; Informix® Vista™; InfoShelf™; Interforum™; I-Spy™; Media360™; Mediation™; MetaCube®, NewEra™; Office Connect™; ON-Bar™; OnLine Dynamic Server™; OnLine/Secure Dynamic Server™; OpenCase®; Orca™; PaVER™; Red Brick® and Design; Red Brick® Data Mine™; Red Brick Decision Server™; Red Brick® Mine Builder™; Red Brick® Decisionscape™; Red Brick® Ready™; Red Brick Systems®; Regency Support®, Rely on Red BrickSM; RISQL®, Server Administrator™; Solution DesignSM; STARindex™; STARjoin™; SuperView®, TARGETindex™; TARGETjoin™; The Data Warehouse Company®; Universal Data Warehouse Blueprint™; Universal Database Components™; Universal Web Connect™; ViewPoint®, Virtual Table Interface™; Visionary™; Web Integration Suite™; XML DataPort™. The Informix logo is registered with the United States Patent and Trademark Office. The DataBlade logo is registered with the United States Patent and Trademark Office.

Documentation Team: Jennifer Leland, Barbara Nomiyama, Tom Noronha, Virginia Panlasigui, Keldyn West

GOVERNMENT LICENSE RIGHTS

Software and documentation acquired by or for the US Government are provided with rights as follows:

(1) if for civilian agency use, with rights as restricted by vendor's standard license, as prescribed in FAR 12.212; (2) if for Dept. of Defense use, with rights as restricted by vendor's standard license, unless superseded by a negotiated vendor license, as prescribed in DFARS 227.7202. Any whole or partial reproduction of software or documentation marked with this legend must reproduce this legend.

Table of Contents

Introduction

In This Introduction	3
Major Features of This Product	3
About This Guide	3
Types of Users	4
Software Dependencies	4
Documentation Conventions	5
Typographical Conventions	5
Icon Conventions	6
Syntax Conventions	7
Additional Documentation	9
Online Manuals	9
Printed Manuals	9
Online Help	10
Documentation Notes, Release Notes, Machine Notes	10
Related Reading	11
Informix Welcomes Your Comments	11

Chapter 1

MaxConnect Overview

In This Chapter	1-3
MaxConnect Functions	1-3
MaxConnect Benefits	1-4
MaxConnect Architecture	1-5
Connection Multiplexing	1-6
Aggregation of Network Messages	1-6
Monitoring and Administering MaxConnect	1-6

Chapter 2

Configuring MaxConnect

In This Chapter	2-3
Configuration Alternatives	2-3
Configuration Overview	2-7
Configuration Without MaxConnect	2-7
Configuration with MaxConnect	2-8
Configuring the Database Server	2-9
Modifying the sqlhosts File	2-9
Changing the INFORMIXSERVER Environment Variable	2-10
Modifying the ONCONFIG File.	2-11
Configuring the MaxConnect Host	2-14
Setting Environment Variables	2-14
Modifying the sqlhosts File	2-15
Specifying Configuration Parameters	2-17
Setting Operating-System Parameters	2-18
Configuring the Client Host	2-18
Modifying the sqlhosts File	2-18
Modifying the sqlhosts Information	2-19
Multiple Database Servers	2-19
Configuring the imcadmin Utility	2-19
Updating the sqlhosts File.	2-19
Specifying an Environment Variable	2-20

Chapter 3

Installing MaxConnect

In This Chapter	3-3
Preparing for MaxConnect	3-3
Running the installimc Script	3-4
Syntax	3-4
Running the configimc Script	3-6
Syntax	3-6
Naming the Configuration File	3-7
Creating the MaxConnect Configuration File	3-8
Creating or Modifying the sqlhosts File	3-9
Performing Manual Tasks to Complete Configuration	3-15
Running the RUN_AS_ROOT.imc Script	3-17

Chapter 4	Administering MaxConnect	
	In This Chapter	4-3
	How MaxConnect Works	4-4
	Listener Thread	4-5
	Sessions	4-5
	Worker Queues and Threads	4-5
	Transports	4-6
	Receive Threads	4-6
	MaxConnect Poll Threads	4-6
	Database Server Poll Threads	4-6
	Starting and Stopping MaxConnect	4-7
	Starting MaxConnect	4-7
	Stopping MaxConnect	4-7
	Using the Informix Server Administrator Utility	4-8
	Using ISA to Administer MaxConnect	4-8
	Using ISA to Monitor MaxConnect	4-8
	Using the imcadmin Utility	4-9
	Using the Command-Line Mode	4-10
	Using Interactive Mode	4-10
	Syntax of the imcadmin Command	4-11
	Monitoring MaxConnect on the Database Server Using the onstat Utility	4-25
	Monitoring Error Messages	4-31
Chapter 5	Tuning Performance	
	In This Chapter	5-3
	Planning and Configuring for Performance	5-3
	Number of Poll Threads	5-4
	Number of Network Protocols	5-4
	Process Priority of MaxConnect	5-6
	Multiplexed Connections Using NETTYPE sqlmux and MaxConnect Together	5-6
	Use of MaxConnect with Wide-Area Networks	5-7
	Monitoring and Tuning Configuration Parameters	5-8
	Number of Sessions in NETTYPE	5-8
	Number of Transports	5-11
	Number of Worker Threads	5-16
	Worker Thread Delay and Packet Aggregation	5-17

Load Balancing Across Multiple Instances	5-23
Establishing Server Groups on UNIX	5-23
Establishing Server Groups on Windows NT	5-24
Providing for MaxConnect Failure	5-25

Appendix A Establishing Server Groups Using regedt32

Index

Introduction

In This Introduction	3
Major Features of This Product.	3
About This Guide	3
Types of Users	4
Software Dependencies	4
Documentation Conventions	5
Typographical Conventions	5
Icon Conventions	6
Comment Icons	6
Feature, Product, and Platform Icons	7
Syntax Conventions	7
Additional Documentation	9
Online Manuals	9
Printed Manuals	9
Online Help	10
Documentation Notes, Release Notes, Machine Notes	10
Related Reading	11
Informix Welcomes Your Comments.	11

In This Introduction

This introduction provides an overview of the information in this guide and describes the conventions it uses.

Major Features of This Product

Informix MaxConnect enables Informix Dynamic Server 2000 to support greatly increased numbers of client connections without degradation of end-user response time. MaxConnect is a new software tier introduced between the database server and clients that transparently funnels multiple client connections onto a smaller number of server connections. A ratio of 100:1 is typical. The database server is freed from managing thousands of client connections, which results in improved response time and decreased CPU cost for the database server. You can use MaxConnect with Informix database servers on UNIX and to support connections from client applications that run on either UNIX or Windows NT.

About This Guide

This guide contains information about Informix MaxConnect, including concepts related to managing client connections and how to install, configure, administer, and tune the performance of MaxConnect.

This section discusses the organization of the guide, the intended audience, and the associated software products that you must have to use MaxConnect.

Types of Users

This guide is for database server administrators and system administrators and is written with the assumption that you have the following background:

- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience with database server administration, operating-system administration, or network administration
- A working knowledge of the architecture and features of Informix Dynamic Server

Software Dependencies

This guide assumes that you are using Informix Dynamic Server 2000 (Version 9.21 or higher).

Documentation Conventions

This section describes the conventions that this guide uses. These conventions make it easier to gather information from this and other volumes in the documentation set.

The following conventions are discussed:

- Typographical conventions
- Icon conventions
- Syntax conventions
- Command-line conventions
- Sample-code conventions

Typographical Conventions

This guide uses the following conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning
KEYWORD	All primary elements in a programming language statement (keywords) appear in uppercase letters in a serif font.
<i>italics</i> <i>italics</i> <i>italics</i>	Within text, new terms and emphasized words appear in italics. Within syntax and code examples, variable values that you are to specify appear in italics.
boldface <i>boldface</i>	Names of program entities (such as classes, events, and tables), environment variables, file and pathnames, and interface elements (such as icons, menu items, and buttons) appear in boldface.
monospace <i>monospace</i>	Information that the product displays and information that you enter appear in a monospace typeface.

(1 of 2)



Convention	Meaning
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.
◆	This symbol indicates the end of product- or platform-specific information.
→	This symbol indicates a menu item. For example, “Choose Tools→Options ” means choose the Options item from the Tools menu.

(2 of 2)




***Tip:** When you are instructed to “enter” characters or to “execute” a command, immediately press RETURN after the entry. When you are instructed to “type” the text or to “press” other keys, no RETURN is required.*

Icon Conventions

Throughout the documentation, you will find text that is identified by several different types of icons. This section describes these icons.



Comment Icons

Comment icons identify three types of information, as the following table describes. This information always appears in italics.

Icon	Label	Description
	<i>Warning:</i>	Identifies paragraphs that contain vital instructions, cautions, or critical information
	<i>Important:</i>	Identifies paragraphs that contain significant information about the feature or operation that is being described
	<i>Tip:</i>	Identifies paragraphs that offer additional details or shortcuts for the functionality that is being described

Feature, Product, and Platform Icons

Feature, product, and platform icons identify paragraphs that contain feature-specific, product-specific, or platform-specific information.

Icon	Description
	Identifies information that is specific to UNIX platforms
	Identifies information that is specific to the Windows NT environment

These icons can apply to an entire section or to one or more paragraphs within a section. If an icon appears next to a section heading, the information that applies to the indicated feature, product, or platform ends at the next heading at the same or higher level. A ♦ symbol indicates the end of feature-, product-, or platform-specific information that appears in one or more paragraphs within a section.

Syntax Conventions

This section describes conventions for syntax diagrams. Each diagram displays the sequences of required and optional keywords, terms, and symbols that are valid in a given statement, command line, or other specification, as in [Figure 1](#).

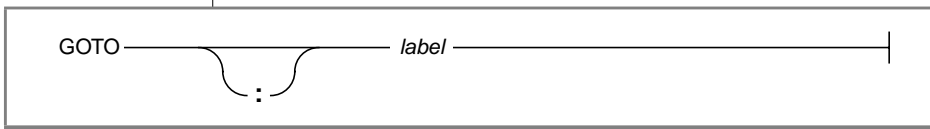


Figure 1
Example of a
Simple Syntax
Diagram

Keep in mind the following rules when you read syntax diagrams in this book:

- To make keywords (like GOTO in [Figure 1](#)) easy to identify, they are shown in UPPERCASE letters, even though you can type them in either uppercase or lowercase letters.
- Terms for which you must supply specific values are in *italics*. In [Figure 1](#), you must replace *label* with an identifier. Below each diagram that contains an italicized term, a table explains what you can substitute for the term.
- All the punctuation and other nonalphabetic characters are literal symbols. In [Figure 1](#), the colon is a literal symbol.
- Each syntax diagram begins at the upper-left corner and ends at the upper-right corner with a vertical terminator. Between these points, any path that does not stop or reverse direction describes a possible form of the statement.

Syntax elements in a path represent terms, keywords, symbols, and segments that can appear in your statement. The path always approaches elements from the left and continues to the right, except in the case of separators in loops. For separators in loops, the path approaches counterclockwise from the right. Unless otherwise noted, at least one blank character separates syntax elements.

Additional Documentation

For additional information, you might want to refer to the following types of documentation:

- Online manuals
- Printed manuals
- Online help
- Error message files
- Documentation notes, release notes, and machine notes
- Related reading

Online Manuals

An Answers OnLine CD that contains Informix manuals in electronic format is provided with your Informix products. You can install the documentation or access it directly from the CD. For information about how to install, read, and print online manuals, see the installation insert that accompanies Answers OnLine. You can obtain the same online manuals on the Web at <http://www.informix.com/answers>.

Printed Manuals

To order printed manuals, call 1-800-331-1763 or send email to moreinfo@informix.com. Please provide the following information when you place your order:

- The documentation that you need
- The quantity that you need
- Your name, address, and telephone number

Online Help

Informix provides online help with each graphical user interface (GUI) that displays information about those interfaces and the functions that they perform. Use the help facilities that each GUI provides to display the online help.

Documentation Notes, Release Notes, Machine Notes

In addition to printed documentation, the following sections describe the online files that supplement the information in this guide. Examine these files before you begin using your database server. They contain vital information about application and performance issues.

The following online files appear in the `$INFORMIXDIR/release/en_us/0333` directory.

Online File	Purpose
ADMINDOC_1.0	The documentation notes file for your version of this guide describes features that are not covered in the guide or that were modified since publication.
IMCREL_1.0	The release notes file describes feature differences from earlier versions of Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds.
IMC_1.0	The machine notes file describes any special actions that you must take to configure and use Informix products on your computer. Machine notes are named for the product described.

You can obtain the same notes and other installation information on the Web at <http://www.informix.com/informix/resource/>.

Related Reading

The following publications provide additional information about the topics that this guide discusses:

- The *Administrator's Guide for Informix Dynamic Server 2000* contains detailed information on how to administer Informix Dynamic Server.
It shows how to install, configure, administer, and use the database server.
- The *Informix Administrator's Reference* provides reference material about Informix Dynamic Server.
It shows the syntax of database server utilities such as **onmode** and **onstat** and gives comprehensive descriptions of configuration parameters, the **sysmasters** tables, and logical-log records.
- The *Performance Guide for Informix Dynamic Server 2000* provides information about how to configure and operate Informix Dynamic Server to improve overall system throughput and to improve the performance of SQL queries.

Informix Welcomes Your Comments

Let us know what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about any corrections or clarifications that you would find useful. Include the following information:

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

Send electronic mail to us at the following address:

`doc@informix.com`

This address is reserved for reporting errors and omissions in our documentation. For immediate help with a technical problem, contact Informix Customer Services.

We appreciate your suggestions.

MaxConnect Overview

In This Chapter	1-3
MaxConnect Functions	1-3
MaxConnect Benefits	1-4
MaxConnect Architecture	1-5
Connection Multiplexing	1-6
Aggregation of Network Messages	1-6
Monitoring and Administering MaxConnect	1-6



In This Chapter

Informix MaxConnect solves the connection scalability problem that arises as the number of connections to a single database server increases.

This chapter introduces the following information:

- [MaxConnect Functions](#)
- [MaxConnect Benefits](#)
- [MaxConnect Architecture](#)
- [Monitoring and Administering MaxConnect](#)

MaxConnect Functions

MaxConnect enables the database server to support an unlimited number of client connections without performance degradation. MaxConnect does not impact the stability of the database server, existing client applications, or legacy functionality.

MaxConnect can be used with Informix database servers on UNIX and supports connections from client applications that run on either UNIX or Windows NT.

MaxConnect can manage large numbers of client-server or server-server connections, from several hundred to tens of thousands. A large number of connections can require a significant percentage of central processing unit (CPU) resources on the database server to manage and can cause degradation of throughput and response time. MaxConnect moves the cost of connection management from the database server to the client or a dedicated computer to improve system scalability, throughput, and flexibility.

MaxConnect Benefits

The primary benefits of MaxConnect are improved scalability, performance, and response time on networks with a medium to large number of client connections. With more than 200 or so user connections, performance improves dramatically. The precise performance improvements realized with MaxConnect vary depending on platform, network hardware, application work load, and any existing bottlenecks in the system.

MaxConnect provides the following benefits:

- Increases the possible number of client connections
It provides support for tens of thousands of client connections by multiplexing connections across a much smaller number of connections to the database server.
- Reduces usage of CPU and system resources on the database server computer by reducing the number of physical connections and poll threads on the database server
- Improves performance (both throughput and response time) by aggregating data packets sent to the database server and thereby reducing the number of data transfers
- Provides transport bridging for client- and server-side systems that have different protocols
- Improves performance of application that are connect-disconnect intensive

MaxConnect can reduce the processing overhead on the database server in the execution of an application that involves many users constantly connecting to or disconnecting from the database server.

MaxConnect is most beneficial with sessions that transfer small amounts of data. Using MaxConnect to transfer large data objects reduces CPU usage on the database server but frequent transfer of large data objects might have some impact on response time. If your goal is to reduce CPU utilization on the database server, you can use MaxConnect for sessions that routinely transfer large objects less than 256 kilobytes. However, if response time is a priority when sessions routinely transfer large objects larger than 256 kilobytes, the sessions should not go through MaxConnect.

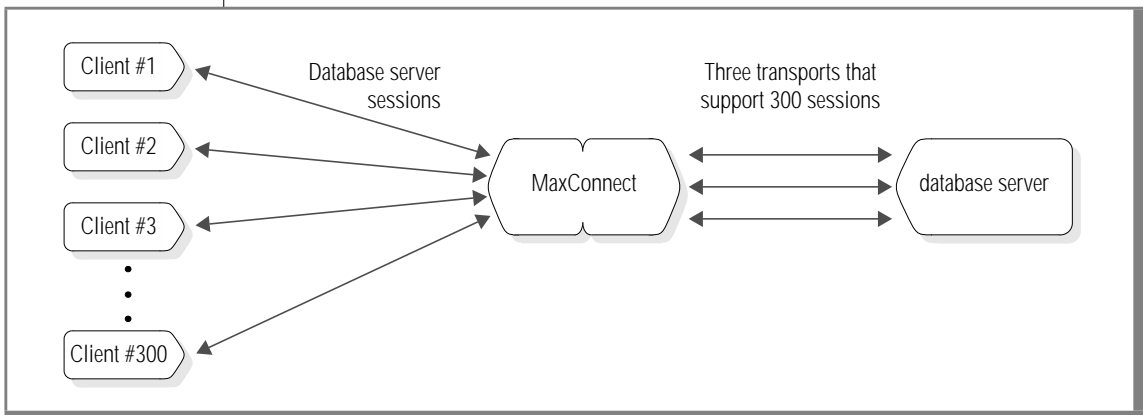
MaxConnect Architecture

MaxConnect is interposed, transparently to the client, between client applications and the database server. Client connections that use MaxConnect are handled in the following manner:

1. MaxConnect listens for connection requests from clients and establishes logical connections that then intercept SQL requests.
2. MaxConnect aggregates these SQL requests into packets and sends the packets to the database server.
3. The database server processes requests and sends packets back to MaxConnect.
4. MaxConnect separates the responses and sends them to the appropriate client application.

Figure 1-1 illustrates a high-level overview of the MaxConnect architecture. In this example 300 client connections are multiplexed over three transports to the database server. The ratio of client sessions to database connections can be more than 100:1.

Figure 1-1
Overview of MaxConnect Architecture



Connection Multiplexing

MaxConnect processes are interposed in place of the standard configuration in which clients send requests directly to the database server. New connections from clients are accepted by the MaxConnect listener thread. Data received from the clients is placed in the data buffers of a worker queue. Each worker queue is associated with a particular worker thread. The worker thread aggregates packets from different clients and sends the aggregated packets to the database server. The MaxConnect receive threads read the aggregated data, de-multiplex this data, and send individual data packets to their corresponding clients.

Aggregation of Network Messages

In typical online transaction processing (OLTP) environments, small data units are exchanged between the client and the database server at a high frequency. MaxConnect aggregates requests from different client sessions to improve performance. Aggregation of messages reduces CPU consumption and improves response time by the database server by reducing system calls.

Monitoring and Administering MaxConnect

You can use the following utilities to start, stop, monitor, and administer MaxConnect:

- The **imcadmin** utility allows you to start and stop MaxConnect, view status information and a variety of statistics, and reset statistics. You can perform any of the functions that the **imcadmin** utility provides from the command line or from an interactive mode.
- The **onstat** utility allows you to monitor MaxConnect statistics on the database server.
- Informix Server Administrator (ISA) allows you to perform all of the functions of the **imcadmin** utility or **onstat** utilities. This web-based interface is used to perform administration tasks on Informix database servers, including those involving the MaxConnect connection server.

Security is regulated in the same manner with MaxConnect as without. Any user trying to connect to the database server will continue to be authenticated. MaxConnect supports all *trusted host* semantics.

For more information on monitoring and administering MaxConnect, refer to [Chapter 4, “Administering MaxConnect.”](#)

Configuring MaxConnect

In This Chapter	2-3
Configuration Alternatives	2-3
Configuration Overview	2-7
Configuration Without MaxConnect	2-7
Configuration with MaxConnect.	2-8
Configuring the Database Server	2-9
Modifying the sqlhosts File.	2-9
Changing the INFORMIXSERVER Environment Variable	2-10
Modifying the ONCONFIG File	2-11
Adding a DBSERVERALIASES entry	2-11
Updating the NETTYPE Configuration Parameter	2-12
Configuring the MaxConnect Host	2-14
Setting Environment Variables	2-14
Modifying the sqlhosts File.	2-15
Specifying Configuration Parameters	2-17
Setting Operating-System Parameters	2-18
Configuring the Client Host	2-18
Modifying the sqlhosts File.	2-18
Modifying the sqlhosts Information.	2-19
Multiple Database Servers	2-19
Configuring the imcadmin Utility.	2-19
Updating the sqlhosts File	2-19
Specifying an Environment Variable	2-20

In This Chapter

This chapter explains the concepts you need to understand to configure MaxConnect on your network. The chapter consists of the following sections:

- [Configuration Alternatives](#)
- [Configuration Overview](#)
- [Configuring the Database Server](#)
- [Configuring the MaxConnect Host](#)
- [Configuring the Client Host](#)
- [Configuring the imcadmin Utility](#)

Configuration Alternatives

You can install MaxConnect on a dedicated computer or on client application computers. Both MaxConnect and the database server must be on UNIX.

If you utilize a high percentage of the CPU on your client application server or if you have only clients and no client application servers in your environment, Informix recommends that you install MaxConnect on a dedicated system as [Figure 2-1 on page 2-4](#) shows.

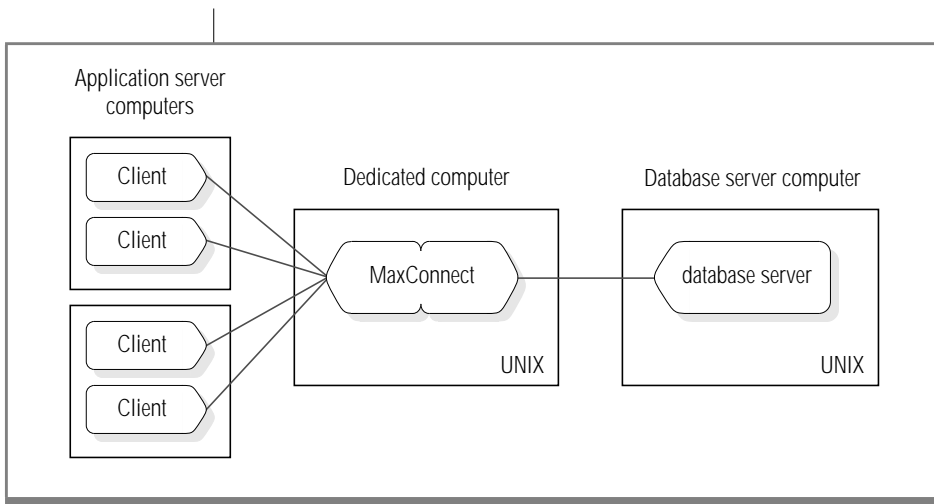


Figure 2-1
*MaxConnect
Installed on a
Dedicated
Computer*

If the CPU utilization on your application server is not high, and if the application server runs on UNIX, Informix recommends that you install MaxConnect on the application servers as [Figure 2-2](#) shows.

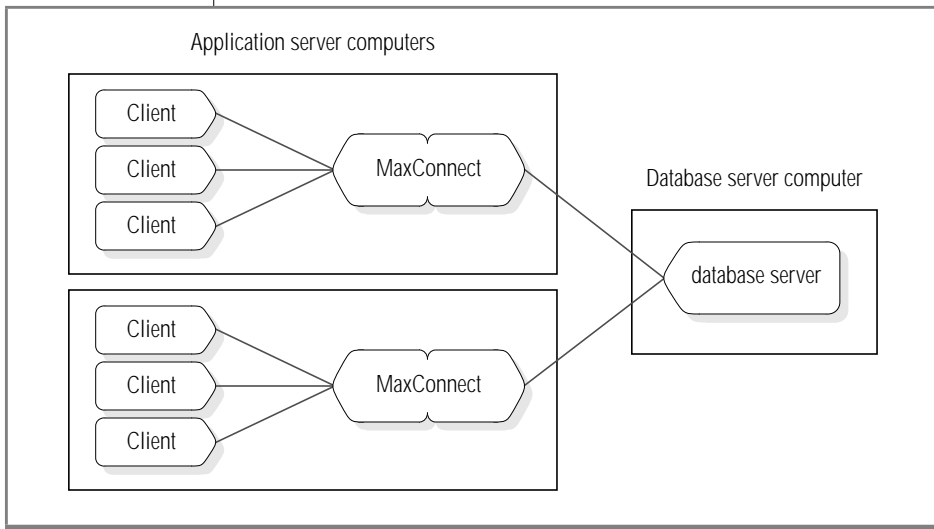
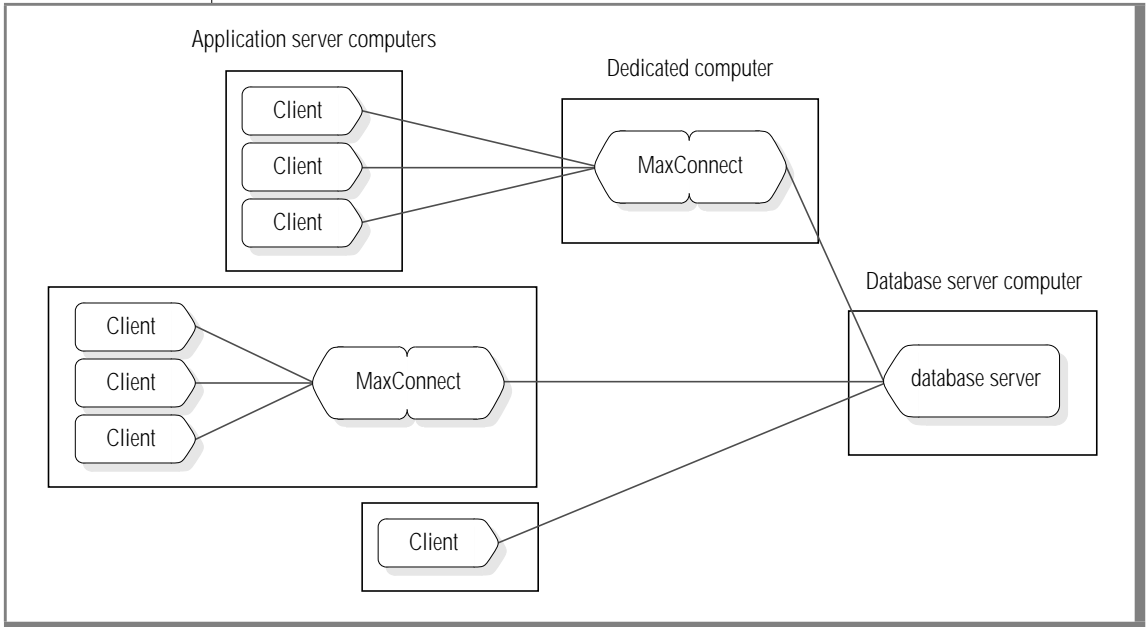


Figure 2-2
*MaxConnect
Installed on
Application
Servers*

You can also choose a combination of these options. You send some application servers through a MaxConnect instance installed on the same computer, send other clients through an instance on a separate computer, and connect other clients directly to the database server. [Figure 2-3](#) illustrates this concept.

Figure 2-3
MaxConnect Installed in Combination



A MaxConnect instance can connect to only one database server. To connect client applications to more than one database server, install multiple instances of MaxConnect, one for each database server as [Figure 2-4](#) illustrates.

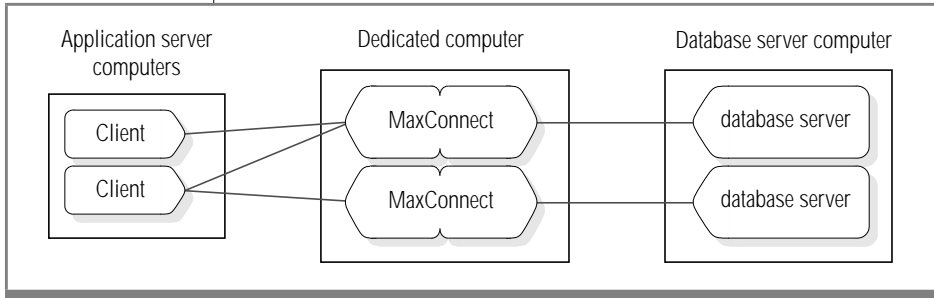


Figure 2-4
Connecting to
Multiple Database
Servers

You can use MaxConnect to manage server-to-server connections, such as those in distributed queries (which allow users to query more than one database server within a single transaction). You can configure the connections to connect through the MaxConnect instance. To use MaxConnect to handle connections between the database servers in both directions, configure two MaxConnect instances, one for each direction. [Figure 2-5](#) illustrates two database servers that connect through MaxConnect.

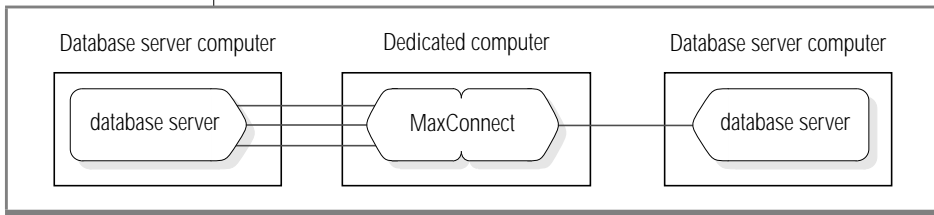


Figure 2-5
Connecting from
Database Server to
Database Server



Tip: You might install MaxConnect on the same computer as the database server on a large hardware-partitionable computer. MaxConnect is not targeted for this configuration and it is not recommended. Even with a symmetric multiprocessor, installing MaxConnect on the same computer as the database server bypasses much of the benefit of the product.

Configuration Overview

To use MaxConnect, you must:

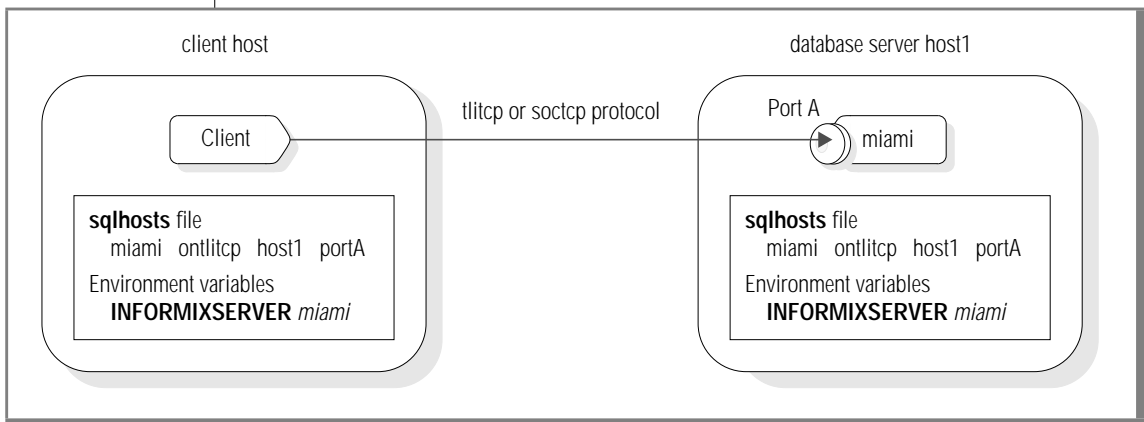
- Make changes to the **sqlhosts** files of the database server and client to route SQL requests through MaxConnect
- Adjust the ONCONFIG file of the database server to add a MaxConnect protocol and reduce the number of poll threads
- On the computer that MaxConnect is installed on, create an **sqlhosts** file for MaxConnect and set MaxConnect environment variables and configuration parameters

Configuration Without MaxConnect

Without MaxConnect, the computers are configured so that the client application connects directly to the database server, as [Figure 2-6](#) shows.

Figure 2-6

Example of System Configuration Without MaxConnect



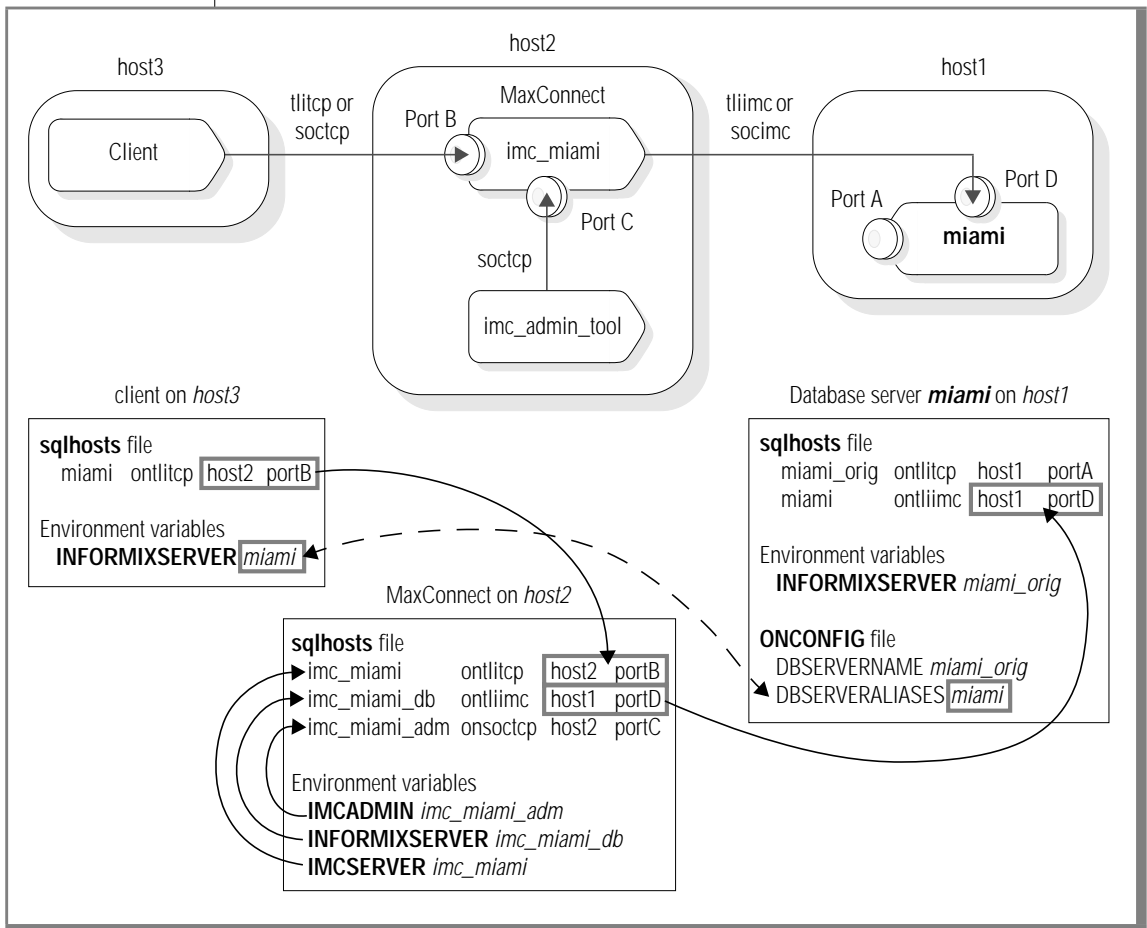
For general information on client/server communications, see the [Administrator's Guide for Informix Dynamic Server 2000](#).

Configuration with MaxConnect

With MaxConnect, the client applications connect to the database server through a MaxConnect instance that might be on the same or a separate computer. Using MaxConnect requires modifications to the configuration of the database server and the client, and a coordinating setup of the MaxConnect host.

[Figure 2-7](#) shows an overview of the network configuration. The configuration of each of the three computers is detailed in the sections that follow.

Figure 2-7
System Configuration with MaxConnect



Important: If MaxConnect is installed on the client application computer, MaxConnect can share the `sqlhosts` file with the client. If the `sqlhosts` file is shared, the value of the `IMCSERVER` environment variable is the same as the `INFORMIXSERVER` environment variable on the client (in this example, `miami`).



Tip: If MaxConnect is installed on the same computer as the client, you can also use the `ipcstr` protocol.

Configuring the Database Server

To use MaxConnect you must make the following changes to the configuration of the database server:

- Modify the **sqlhosts** file
- Change the **INFORMIXSERVER** environment variable
- Modify the ONCONFIG file
 - Add a DBSERVERALIASES entry
 - Update the NETTYPE configuration parameter

The installation script reminds you to complete these tasks manually after installation is complete.

Modifying the sqlhosts File

The **sqlhosts** file contains information that the database server uses to start listener threads for the clients.

The following example shows the original entry in the **sqlhosts** file.

	dbservername	nettype	hostname	servicename
Original entry	miami	ontlitcp	host1	portA

Make the following changes to the **sqlhosts** file:

- Modify the existing entry by changing the **dbservername**.
This listener thread is used by administration utilities (such as **dbaccess**, **oncheck**, and **onspaces**) to connect to the database server.
- Add a new entry that specifies a MaxConnect protocol that is compatible with the protocol that your operating system uses. For instance, if your computer use the sockets protocol, add **onsocimc**; if your computer uses TLI/TCP, add **ontliimc**.

This listener thread is used by MaxConnect to connect to the database server.

	dbservername	nettype	hostname	servicename
Modified original entry	miami_orig	ontlitcp	host1	portA
Additional entry	miami	ontliimc	host1	portD

You can choose to configure just one listener thread for MaxConnect if you want to avoid the overhead on the database server of maintaining two. For more information, see [“Number of Network Protocols” on page 5-4](#).

Tip: If you have multiple database servers on the same host that communicate with each other, create separate **sqlhosts** files for each database server in order to simplify the initial configuration process for MaxConnect.

Important: The client application and database server must have separate **sqlhosts** files when MaxConnect is used.

Changing the INFORMIXSERVER Environment Variable

Change the setting of the **INFORMIXSERVER** environment variable to the value of the original **sqlhosts** file entry that you modified, as in the example above. In the example, the new value of **INFORMIXSERVER** is *miami_orig*.



Modifying the ONCONFIG File

You must make the following modifications to the ONCONFIG file:

- Add a DBSERVERALIASES entry.
- Modify the original NETTYPE entry.
- Add a NETTYPE entry.

Adding a DBSERVERALIASES entry

Add a DBSERVERALIASES entry to the ONCONFIG file to configure a listener thread for MaxConnect instances. In the example above, the new DBSERVERALIASES entry is **miami**, which corresponds to the entry that you added to the **sqlhosts** file for MaxConnect.

For example, the following table shows the original entry.

	ONCONFIG File Entry	Value
Original entry	DBSERVERNAME	miami

The next table shows the modified ONCONFIG file.

	ONCONFIG File Entry	Value
Modified original entry	DBSERVERNAME	miami_orig
Additional entry	DBSERVERALIASES	miami



Important: The value of **INFORMIXSERVER** on the client host must match the **DBSERVERNAME** or a **DBSERVERALIASES** entry in the **ONCONFIG** file of the database server. Because you used the **dbservername** from the entry that client hosts use to connect to the database server for the **MaxConnect** entry, you do not need to change the value of **INFORMIXSERVER** on the client.

Updating the NETTYPE Configuration Parameter

The following table shows an example of the original NETTYPE entry to the **onconfig** file.

	Parameter	Protocol	# of Poll Threads	Total # of Sessions	Virtual Processor Class
Original entry	NETTYPE	ontlitcp	4	500	CPU

Modify this entry and add a new entry for the MaxConnect protocol. If all client sessions are connected through MaxConnect, reduce the value in the field for the number of poll threads to 1 and reduce the total number of sessions to 100 or so, as the following table shows.

	Parameter	Protocol	# of Poll Threads	Total # of Sessions	Virtual Processor Class
Modified original entry	NETTYPE	ontlitcp	1	100	CPU
Additional entry	NETTYPE	ontliimc	1	2000	CPU

To run MaxConnect efficiently, choose appropriate values for each option of the new NETTYPE configuration parameter. The recommended values are discussed in the following table.

Option	Recommended Value
Protocol	Choose a protocol of one of the MaxConnect types. For instance, change tlitcp to tliimc and change soctcp to socimc. The new protocols encompass the old protocols. For instance, a database server that uses the ontliimc protocol recognizes connections that use ontlitcp. MaxConnect transparently supports multiplexed connections that use the special NETTYPE parameter that has the value of sqlmux.
Number of Poll Threads	A value of 1 or 2 is recommended for both the original and the new entry if the majority of clients are connecting through MaxConnect. If a majority of clients continue to connect directly to the database server, you might need more poll threads.
Total Number of Sessions	This number indicates (but does not limit) the total number of concurrent user sessions connected to the database server, either directly or through instances of MaxConnect. Because the database server uses this number to tune internal data structures, it should be close to the actual number of users. For information on performance tuning this value, see “Number of Sessions in NETTYPE” on page 5-8 .
Virtual Processor Class	This field defines the class of virtual processors that run the poll threads. For performance reasons, Informix recommends that you use the CPU VP type rather than the NET type.

Configuring the MaxConnect Host

The configuration script that you run as part of the installation process performs the tasks necessary to configure the MaxConnect host computer. These tasks are:

- setting the environment variables.
- creating or modifying the **sqlhosts** file.
- creating a MaxConnect configuration file that contains four MaxConnect-specific configuration parameters.
- setting operating-system parameters.

The configuration script, described in [Chapter 3, “Installing MaxConnect,”](#) either prompts you for input or uses default settings for the values of the environment variables, **sqlhosts** file entries, and configuration parameters. You can modify any of the settings after the script completes.

For more information about environment variables, **sqlhosts** files, and configuration parameters, see the [Administrator’s Guide for Informix Dynamic Server 2000](#).

Setting Environment Variables

There are six environment variables that pertain to MaxConnect. Three of these environment variables are global to the Informix environment: **INFORMIXDIR**, **INFORMIXSQLHOSTS**, and **INFORMIXSERVER**. Three are specific to MaxConnect: **IMCONFIG**, **IMCSERVER**, and **IMCADMIN**. The following table describes each of these environment variables.

Environment Variable	Description
INFORMIXDIR	Specifies the directory where MaxConnect is installed. This directory is the base for the default values of most of the following environment variables. You must specify a value for INFORMIXDIR .
INFORMIXSQLHOSTS	Specifies the location of the sqlhosts file that contains information about each MaxConnect instance and the database server to which it is connected. The default value is \$INFORMIXDIR/etc/sqlhosts .
INFORMIXSERVER	Selects an entry in the sqlhosts file to identify the database server to which MaxConnect connects. You must specify a value for INFORMIXSERVER .
IMCCONFIG	Identifies the location of the MaxConnect configuration file. The default location and name is \$INFORMIXDIR/etc/IMCconfig .
IMCSERVER	Identifies the dbservername entry in the sqlhosts file that clients use to connect to MaxConnect. You must specify a value for IMCSERVER .
IMCADMIN	Used by the imcadmin tool to access the sqlhosts file to determine how to connect to MaxConnect, and used by MaxConnect to determine where to establish the imcadmin listener port. You must specify a value for IMCADMIN .

You can set these environment variables in your shell. To configure and administer MaxConnect using the Informix Server Administrator (ISA) tool, set environment variables for MaxConnect in the ISA configuration file. For more information, see online help for ISA.

Modifying the sqlhosts File

When you install MaxConnect on a separate computer, the installation script creates an **sqlhosts** file that contains Informix client-server connectivity information for the computer where MaxConnect runs.

When you install MaxConnect on the client computer, the installation script modifies the existing **sqlhosts** file, and MaxConnect and the client application share the same file. If you plan to modify the **sqlhosts** file after installation, Informix recommends that you select the option of creating a separate **sqlhosts** files for MaxConnect and the clients when the installation script prompts you. The following example shows a MaxConnect **sqlhosts** file:

```
# on host2: /informix/imc/etc/sqlhosts:

# these entries used by IMC

# used by the client to connect to IMC
imc_miami      ontlitcp      host2          7101

# used by the IMC Admin thread
imc_miami_adm  onsoctcp      host2          7102

# used by IMC to connect to the DB Server
imc_miami_db   ontliimc      host1          7104
```

To install multiple instances connecting to the same database server, include entries for each MaxConnect instance and its associated **imcadmin** utility.

The following table shows an example of the entries in the **sqlhosts** file.

dbservername	nettype	hostname	servicename
imc_miami_db	ontliimc	host1	dbservice
imc_server1	ontlitcp	host2	imcservice1
imc_server2	ontlitcp	host2	imcservice2
imc_admin1	ontlitcp	host2	imcadminservice1
imc_admin2	ontlitcp	host2	imcadminservice2

Specifying Configuration Parameters

The installation script creates a MaxConnect configuration file with the following four configuration parameters:

- IMCLOG
- IMCWORKERTHREADS
- IMCWORKERDELAY
- IMCTRANSSPORTS

The default location of this file is `$INFORMIXDIR/etc/IMCconfig`. The parameters are described in the following table.

Figure 2-8
Configuration Parameters

Configuration Parameter	Description
IMCLOG	<p>Defines the location of the log file on the MaxConnect host to which all MaxConnect messages, both errors and warnings, are written. The installation script prompts you for the initial value.</p> <p>The default value is <code>imc.log</code> in the same directory as the MaxConnect executable.</p>
IMCWORKERTHREADS	<p>Specifies the number of worker threads. One thread is enough for most work loads. The installation script uses the default value.</p> <p>The default and minimum value is 1; maximum is 64.</p>
IMCWORKERDELAY	<p>Specifies how long a worker thread waits for messages to aggregate before performing a send operation. This parameter can improve aggregation of messages and thereby reduce CPU consumption on the database server. However, it can have a great effect on response times and should be modified with caution. The installation script uses the default value.</p> <p>The default and minimum value is 0; maximum is 100,000 microseconds.</p>
IMCTRANSSPORTS	<p>Specifies the number of transports established between MaxConnect and the database server. They are all established when the first user connects to MaxConnect. The installation script uses a ratio of 1 transport per 100 users for the initial value of IMCTRANSSPORTS and prompts you for the number of users. You can later adjust the number of transports depending on the amount of throughput on your system.</p> <p>The default value for this parameter is 2; minimum is 1; maximum is 64.</p>

The following example shows a sample configuration file for MaxConnect:

```
IMCLOG /informix/imc/imc.log
IMCWORKERTHREADS 1
IMCWORKERDELAY 0
IMCTRANSPTS 2
```

For information on tuning these parameters, see [Chapter 5, “Tuning Performance.”](#)

Setting Operating-System Parameters

You might need to adjust operating-system parameters on the MaxConnect host. For information on your operating system, refer to the machine notes file. For the location of this file, see “[Documentation Notes, Release Notes, Machine Notes](#)” on [page 10](#) of the Introduction.

Configuring the Client Host

To configure the client host computer so applications connect to MaxConnect rather than directly to the database server, modify the **sqlhosts** information. MaxConnect is transparent to client applications.

For illustrations of the examples in the following sections, see [Figure 2-6 on page 2-7](#) and [Figure 2-7 on page 2-8](#).

UNIX

Modifying the sqlhosts File

For clients on UNIX, modify the **sqlhosts** file that the client application accesses to change the hostname and servicename to the MaxConnect instance that will handle the connection. If MaxConnect is installed on the same computer as the client applications, the installation script makes the necessary modifications to the **sqlhosts** file.

For example, change the following **sqlhosts** file entry:

```
miami      ontlitcp   host1      portA
```

To the following entry:

```
miami      ontlitcp   host2      portD
```

WIN NT

In this example, host1 is the database server computer and host2 is the MaxConnect computer.

Modifying the sqlhosts Information

For clients on Windows NT, use the **setnet32** tool to change the entries for host and service from the hostname and service name of the database server to the name and service of the MaxConnect instance.

Multiple Database Servers

If your client connects to more than one database server through MaxConnect, add a new entry for each of the MaxConnect instances to which it connects.

Configuring the imcadmin Utility

You must also configure the **imcadmin** utility, which is used to monitor MaxConnect instances. For more information, see [“Using the imcadmin Utility” on page 4-9](#).

Updating the sqlhosts File

The installation script asks you for a port number or service name for the **imcadmin** utility and reminds you to add it to the `/etc/services` file. It then adds an entry to the **sqlhosts** file.

Add an additional entry to the **sqlhosts** file for each MaxConnect instance that you want to monitor using this **imcadmin** utility, as the following example shows:

```
imc_miami_adm onsoctcp host2 portC
imc_tucson_adm onsoctcp host5 portC
```

Specifying an Environment Variable

Specify an environment variable for the **imcadmin** utility as the following example shows:

```
setenv IMCADMIN imc_miami_adm
```

This environment variable specifies a MaxConnect instance on the local computer. Configure an **imcadmin** utility for each MaxConnect instance. You can use each **imcadmin** utility to start and stop the MaxConnect instance that the **IMCADMIN** environment variable specifies.

Installing MaxConnect

In This Chapter	3-3
Preparing for MaxConnect	3-3
Running the installimc Script	3-4
Syntax	3-4
Running the configimc Script	3-6
Syntax	3-6
Naming the Configuration File	3-7
Creating the MaxConnect Configuration File	3-8
Creating or Modifying the sqlhosts File	3-9
Creating a New sqlhosts File	3-9
Modifying an Existing sqlhosts File	3-13
Performing Manual Tasks to Complete Configuration	3-15
Running the RUN_AS_ROOT.imc Script	3-17



In This Chapter

This chapter describes how to install MaxConnect and consists of the following sections:

- [Preparing for MaxConnect](#)
- [Running the installimc Script](#)
- [Running the configimc Script](#)
- [Running the RUN_AS_ROOT.imc Script](#)

Preparing for MaxConnect

MaxConnect product files are contained in a file on the distribution media called **imc.tar** or **imc.cpio**.

Before you install MaxConnect, take the following steps

1. Log on as user **informix**.
2. Create a directory to hold the MaxConnect product files.
3. Copy the **imc.tar** file to the directory you just created.
4. Enter the appropriate **cpio**, **tar**, or other load command listed on the serial number keycard to transfer software from the media to the current directory:

```
tar xvf imc.tar
cpio -ivdBum < imc.cpio
```

5. Set the **INFORMIXDIR** environment variable to the name of your installation directory as in the following example:

```
setenv INFORMIXDIR /informix/imc
```

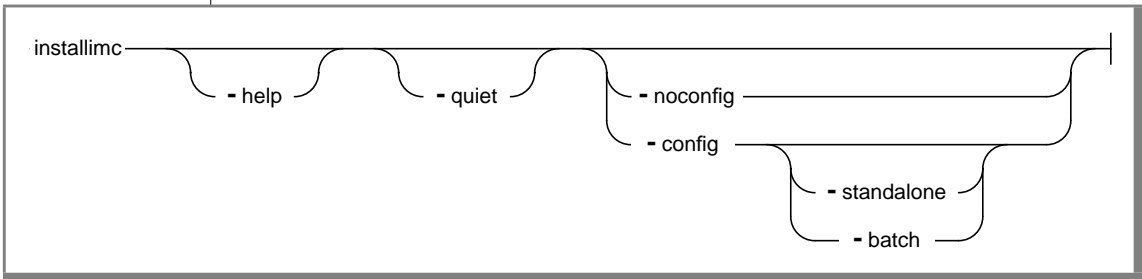


Tip: To uninstall MaxConnect, remove the installation directory. If you have configured the database server, MaxConnect, and the client computers, restore the **sqlhosts** files and environment variable settings to their original state. The **configimc** script makes a backup copy of the **sqlhosts** file before it modifies it on the MaxConnect host.

Running the installimc Script

From the installation directory, as user **informix**, start the **installimc** script by entering the **installimc** command.

Syntax



Element	Short Form	Purpose
-help	-h	Prints the options for this command.
-quiet	-q	Reduces explanatory messages displayed while the configuration script is running. The configimc -quiet command has the same effect.
-noconfig		Completes the installation script process and ends without calling the configuration script.
-config		Completes the installation script process and calls the configuration script.
-standalone	-s	Forces the creation of new sqlhosts file entries for MaxConnect, rather than sharing entries with a client application.
-batch	-b	Configuration script automatically uses the default values and exits at the first unknown value. Use this option when you have successfully installed other MaxConnect instances and have verified that the default values are correct.

In response to the `installimc` command, the script displays the following installation script requirements:

```
Installation Script
```

```
Installation Script Requirements:
```

- A user "informix" and a group "informix" must be known to the system
- The product source files must have been loaded by user informix
- This installation procedure must be run by user informix.

```
This script will change the owner, group, and mode of  
many of the files of this package in this directory.
```

```
Press RETURN to continue,  
or the interrupt key (usually CTRL-C or DEL) to abort.
```

At the following prompts, enter the serial number and serial number key that can be found on the card enclosed with the product shipment:

```
Enter your serial number (for example, INF#X999999) >  
INF#X999999 <RETURN>  
Enter your serial number KEY (uppercase letters only) >  
ABCDEF <RETURN>
```

The script then displays the following warning:

```
WARNING!
```

```
This software, and its authorized use and number of users, are subject to  
the applicable license agreement with Informix Software, Inc. If the number of  
users exceeds the licensed number, the excess users may be prevented from using  
the software. UNAUTHORIZED USE OR COPYING MAY SUBJECT YOU AND YOUR COMPANY TO  
SEVERE CIVIL AND CRIMINAL LIABILITIES.
```

```
Press RETURN to continue,  
or the interrupt key (usually CTRL-C or DEL) to abort.
```

Press RETURN to continue. The script displays the following message that it is installing directories that contain the MaxConnect product files:

```
Installing directory .  
Installing directory bin  
Installing directory etc  
...
```

When the directories are installed, the script informs you that you must run the script **RUN_AS_ROOT.imc**. You can choose to proceed from the installation script directly to the configuration script and run the **RUN_AS_ROOT.imc** script after, or you can change to user **root** and run it after the **installimc** script and then run the **configimc** script.

```
*****  
To complete the installation of Informix Connection Server,  
run /work/imc/RUN_AS_ROOT.imc as root.  
*****
```

```
Informix user portion of installation of Informix Connection Server complete.
```

Running the configmc Script

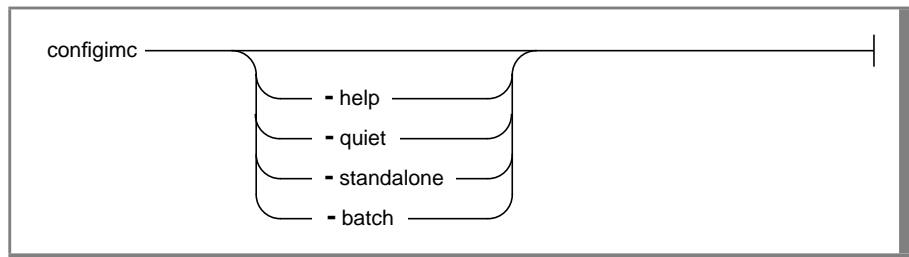
At the end of the installation, the installation script asks whether you want to proceed with the configuration phase of the installation:

```
Informix Connection Server requires a new configuration file  
and changes in the sqlhosts file!
```

```
Do you want to proceed now with this configuration [Y] ? <RETURN>
```

Press **RETURN** or **Y** to start the **configimc** configuration script, which configures the MaxConnect host. If you interrupt this script at any point after it starts, you can restart it by entering **bin/configimc** from the installation directory.

Syntax



Element	Short Form	Purpose
-help	-h	Prints the options for this command.
-quiet	-q	Reduces explanatory messages displayed while the configuration script is running. The configmc -quiet command has the same effect.
-standalone	-s	Forces the creation of a new sqlhosts file for MaxConnect rather than modifying an existing file.
-batch	-b	Runs the installation script automatically using the default values and stops at the first unknown value. Use this option when you have completed previous installations and have verified that the default values are correct.

After you indicate that you want to proceed with the configuration phase, or after you enter the **bin/configmc** command, the script displays the following message:

```
Configuration Script
```

```
Configuration Script Requirements:
```

- A user "informix" and a group "informix" must be known to the system.
- The product source files must have been loaded by user informix and installed with the "installmc" script.
- This configuration procedure must be run by user informix.

```
You can interrupt his script at any point and rerun later.
The -quiet option will generate fewer messages.
```

```
Press RETURN to continue <RETURN>
```

Naming the Configuration File

The configuration script now asks you to name the configuration file:

```
MaxConnect requires a new configuration file,
determined by the IMCCONFIG environment variable.
```

```
$IMCCONFIG is not set Defaulting to $INFORMIXDIR/etc/IMCconfig
This file does not exist and will be created.
```

```
Press Return to confirm [ /informix/imc/etc/IMCconfig ]
or type in new IMCCONFIG: <RETURN>
```

Press RETURN to accept the name of the configuration file or enter a new name. If you enter a new name, enter the complete pathname rather than a filename.

The script informs you that it is proceeding with the designated name:

```
Proceeding with /informix/imc/etc/IMCconfig.  
You can rename this file later if you also update the IMCCONFIG environment  
variable.
```

Creating the MaxConnect Configuration File

The configuration script creates a file that contains the four configuration parameters that MaxConnect requires. The script uses the default value for two of these and asks for input for IMCLOG and IMCTransports. For more information, see [“Specifying Configuration Parameters” on page 2-18](#).

The configuration script asks you to name the MaxConnect log file:

```
This script will now collect information for the new configuration file:
```

```
The IMCLOG parameter contains the pathname of the MaxConnect log file.
```

```
If this pathname does not start with '/', it will be relative  
to the current directory when starting MaxConnect.
```

```
Press Return to confirm [ imc.log ]  
or type in new pathname: <RETURN>
```

Press RETURN to accept a log name relative to the current directory or enter a new pathname.

The script asks you to specify how many users will connect concurrently to this instance of MaxConnect:

```
What is the maximum number of users that will connect  
to this MaxConnect instance [ 200 ] ? 400 <RETURN>
```

The script uses the value that you enter to derive the initial value of the IMCTransports parameter of the configuration file at a ratio of 100 users to 1 transport. If you enter 400, the IMCTransports parameter is set to 4. You can later tune the number of transports to improve performance.

Creating or Modifying the sqlhosts File

MaxConnect requires three entries in the **sqlhosts** file of the host computer. These three entries are used by the client to connect to MaxConnect (to its listener thread), by the **imcadmin** utility to connect to MaxConnect, and by MaxConnect to connect to the database server. For more information, see [“Configuring the imcadmin Utility” on page 2-20](#).

The script checks whether Informix client products are already installed. If no Informix client products are installed, you see this message:

```
MaxConnect also requires an sqlhosts file,  
determined by the INFORMIXSQLHOSTS environment variable.
```

```
Looking for installed Informix Client products...  
No installed Client product has been detected.
```

In this case, the script creates a new **sqlhosts** file.

If the script finds Informix client products (for instance if you are installing on a client application computer), you are asked whether you would like to share the existing file:

```
Looking for installed Informix client products...  
  Informix Client toolkit 2.3.UC1 ... has been detected.  
MaxConnect can share 'sqlhosts' entries with this product.  
Would you like to do so [Y]
```

If you specify **no**, the script creates a new **sqlhosts** file as described in [“Creating a New sqlhosts File” on page 3-9](#). In this case you must modify the client **sqlhosts** file manually after the configuration script ends.

If you specify **yes** or enter **RETURN**, the script modifies the existing **sqlhosts** file as described in [“Modifying an Existing sqlhosts File” on page 3-13](#). In this case, the script makes all modifications necessary and you do not have to modify the client entries in the shared **sqlhosts** file.

Creating a New sqlhosts File

If you are installing on a computer with no Informix client products or if you specify **no** when asked if you want MaxConnect to share an existing **sqlhosts** file, then the configuration script creates an **sqlhosts** file. This is called a standalone configuration. You are prompted to accept the default location of the new **sqlhosts** file or enter a new location.



Tip: The script also uses the **sqlhosts** file location specified in this step for the **INFORMIXSQLHOSTS** environment variable.

```
$INFORMIXSQLHOSTS is not set. Defaulting to $INFORMIXDIR/etc/sqlhosts.  
This file does not exist and will be created.
```

```
Press Return to confirm [ /informix/imc/etc/sqlhosts ]  
or enter new INFORMIXSQLHOSTS: <RETURN>
```

```
*****  
*****
```

```
Proceeding as a standalone configuration
```

If you are performing a new installation, **INFORMIXSERVER** is generally not set. You are prompted to enter the value for the **INFORMIXSERVER** environment variable on the client, as follows:

```
$INFORMIXSERVER is not set.
```

```
Enter CLIENT $INFORMIXSERVER: miami <RETURN>
```

The script uses the string that you enter to generate meaningful names for entries in the new **sqlhosts** file for the listener thread, admin thread, and database server connections.

The script prompts you for input that it uses to create these entries, as follows:

```
** Configuring the MaxConnect listener thread,  
** which handles incoming connections from client processes.
```

```
Supported protocols are onsoctcp, ontlitcp and onipcstr  
Press RETURN to confirm or enter new protocol [ ontlitcp ]: <RETURN>
```

The list of supported protocols that the script displays depends on your platform. In general, enter the protocol that the clients use to connect to the database server without MaxConnect.

If you are installing MaxConnect on an application computer or if clients connect locally, specify the **onipcstr** protocol for maximum efficiency. If you specify **onipcstr** as the protocol for the listener thread, the prompts that follow ask for a stream-pipe name instead of a hostname and service name.

```
Enter port number (or service name) [ 7103 ]: imclsn<RETURN>  
***Service imclsn is not listed in /etc/services file.
```

You will be reminded at the end of the configuration script to manually update the `/etc/services` file with the new entry.

```
Press RETURN to confirm creation or type in new value.  
Enter port number (or service name) [imc1sn]: <RETURN>  
Enter hostame (or IP address) [ host2 ]: <RETURN>
```

Enter the name of the computer on which you are installing this MaxConnect instance. The script informs you that it has added the `sqlhosts` entry for the MaxConnect listener thread:

```
+ added imc_miami entry
```

The script prompts you for the port number for the MaxConnect admin thread:

```
** Configuring the MaxConnect admin thread,  
** which handles requests from the imcadmin utility.  
  
Enter port number (or service name) [ 7104 ]: <RETURN>  
+ added imc_miami_adm entry
```

The script prompts you for the protocol, port number, and hostname of the database server. The list of supported protocols varies with the platform and on certain platforms the prompts for port number or hostname might not appear.

```
** Configuring MaxConnect for communicating with the database server.  
  
Supported protocols are onsocimc, ontliimc  
Press RETURN to confirm or enter new protocol [ ontliimc ]: <RETURN>  
  
Type in the port number (or service name)for the database listener.  
Enter port number (or service name): portD <RETURN>  
  
Type in the hostname (or IP address for the database listener.  
Enter hostname (or IP address): host1 <RETURN>
```

The script informs you that it has added the `sqlhosts` entry for communication between MaxConnect and the database server:

```
+ added imc_miami_db entry
```

After you specify the three **sqlhosts** entries, the script informs you that it is ready to generate the **sqlhosts** file and displays the entries you specified:

```
We're now ready to generate the INFORMIXSQLHOSTS file
(/informix/imc/etc/sqlhosts)
with these changes:
0a1,11
>
> # new MaxConnect section added on 02/24/00
> # entry used by the MaxConnect listener thread
> imc_miami      ontlitcp      host2      imclsn
>
> # entry used by the MaxConnect Admin thread
> imc_miami_adm  onsoctcp      host2      7104
>
> # entry used to connect to the database server
> imc_miami_db   ontliimc      host1      portD
```

After it displays the specified **sqlhosts** entries, the script prompts you to confirm that you want to generate the **sqlhosts** file. If you accept, the script sends the following messages that indicate that the **sqlhosts** file and configuration file have been generated:

```
Do you want to proceed [ Y ] ? <RETURN>
```

```
The /informix/imc/etc/sqlhosts file has been generated.
The /informix/imc/etc/IMCconfig file has been generated.
```

```
Configuration of MaxConnect successfully completed.
```

```
Please consult the MaxConnect Release Notes.
```

The script proceeds to describe further installation tasks, as discussed in [“Performing Manual Tasks to Complete Configuration” on page 3-15](#).

Modifying an Existing sqlhosts File

If you specify **yes** when asked if you want MaxConnect to share an existing **sqlhosts** file, then the configuration script modifies the existing **sqlhosts** file and displays the following dialog:

```
$INFORMIXSQLHOSTS is not set. Defaulting to $INFORMIXDIR/etc/sqlhosts.  
This file exists and a backup copy will be made before modifying it.
```

```
Press RETURN to confirm [/informix/imc/etc/sqlhosts]  
or enter new INFORMIXSQLHOSTS: <RETURN>
```

```
Proceeding with /informix/imc/etc/sqlhosts.
```

The script displays the setting of the **INFORMIXSERVER** environment variable on the MaxConnect host. If you are performing a new installation, **INFORMIXSERVER** is generally not set. If you are installing on a client host, **INFORMIXSERVER** is set to the database server to which the client connects.

```
INFORMIXSERVER is not set.
```

```
Enter sqlhosts entry to modify: miami <RETURN>
```

The script uses this string to generate meaningful names for entries in the existing **sqlhosts** file for the listener thread, admin thread, and database server connections.

The configuration script prompts you for information that it uses to modify the **sqlhosts** file. It modifies the entry for the client to connect to MaxConnect and creates two new entries for the **imcadmin** utility to connect to the MaxConnect instance that you are installing and for MaxConnect to connect to the database server.

```
** Configuring the MaxConnect listener thread,  
** which handles incoming connections from client processes.  
+modified miami entry
```

```
**Configuring the MaxConnect admin thread,  
** which handles requests from the imcadmin utility.  
Enter port number (or service name) [7104]: <RETURN>  
+added imc_miami_adm entry
```

```
** Configuring MaxConnect for communicating with the database server.  
+added imc_miami_db entry
```

The configuration script automatically generates the **sqlhosts** file for the MaxConnect host, as follows:

```
The script is now ready to generate the INFORMIXSQLHOSTS file
( /informix/imc/etc/sqlhosts )
with these changes:
6c6,17
< miami ontlitcp  host1  portA  m=1
----
> # this entry used by the existing client
> # and the MaxConnect listener thread
> miami ontlitcp  host2 imc_p_portA  m=1
>
> # new MaxConnect section added on 02/24/00
> # entry used by the IMC Admin thread
> imc_miami_adm  onsoctcp  host2 7104
>
> # entry used to connect to the database server
> imc_miami_db  ontliimc  host1  portA
>
> # end of new MaxConnect section ( 02/24/00 )

Do you want to proceed [ Y ] ? <RETURN>

Backing up existing /informix/imc/etc/sqlhosts file
into /informix/imc/etc/sqlhosts.bak.0224.19596

The /informix/imc/etc/sqlhosts file has been generated.

Backing up existing /informix/imc/etc/IMCconfig file
into /informix/imc/etc/IMCconfig.bak.0224.19596

The /informix/imc/etc/IMCconfig file has been generated.

Configuration of MaxConnect successfully completed.

Please consult the MaxConnect Release Notes.
```

Performing Manual Tasks to Complete Configuration

After the configuration script informs you that it has prepared the **sqlhosts** file and the configuration file, it lists the following steps that are not under the control of a script and that you need to complete manually:

```
*****  
***   IMPORTANT NOTES   ***  
*****
```

There are several manual steps that you need to execute before you can run MaxConnect!
These steps are described in the imc_0224.steps.txt file.

Do you want to look at this file now [Y] ? <RETURN>

```
*****  
***   MANUAL STEPS   ***  
*****
```

If you installed MaxConnect in a standalone configuration, make the following modifications in the client environment:

In the Client environment, change the miami sqlhosts entry as follows:
the 3rd field matches the MaxConnect listener hostname (host2) and
the 4th field matches the MaxConnect listener port number/service name (imclsn)

WIN NT

For clients on Windows NT, use the **setnet32** tool to change the entries for host and service from the hostname and service name of the database server to the name and service of the MaxConnect instance. ♦

Whether you installed MaxConnect on a standalone computer or on a computer with a client, make the following modifications in the database server and MaxConnect environments:

In the database server environment (host1):
Update the sqlhosts and onconfig files according to
the Guide to Informix MaxConnect.

For more information, see “Configuring the Database Server” on page 2-10.

In the MaxConnect environment,
Verify that the /etc/services file is updated with the following entries:

Use a text editor such as **vi** if you need to modify these entries to the **/etc/services** file:

```
imc_p_portA    10007/tcp    # added by MaxConnect configuration script (02/24/00)  
imc_p_7104    10008/tcp    # added by MaxConnect configuration script (02/24/00)  
Verify that the following environment variables are set as described below:  
INFORMIXDIR, INFORMIXSQLHOSTS, INFORMIXSERVER,  
IMCSERVER, IMCADMIN, and IMCCONFIG.
```

The correct settings for these environment variables as determined by the values you entered during configuration are displayed as follows:

```
With Bourne/Korn shell:
export INFORMIXDIR=/informix/imc
export INFORMIXSQLHOSTS=/informix/imc/etc/sqlhosts
export INFORMIXSERVER=imc_miami_db
export IMCSERVER=imc_miami
export IMCADMIN=imc_miami_adm
export IMCCONFIG=/informix/imc/etc/IMCconfig
export PATH=$INFORMIXDIR/bin:$PATH
```

```
With C shell:
setenv INFORMIXDIR /informix/imc
setenv INFORMIXSQLHOSTS /informix/imc2/etc/sqlhosts
setenv INFORMIXSERVER imc_miami_db
setenv IMCSERVER imc_miami
setenv IMCADMIN imc_miami_adm
setenv IMCCONFIG /informix/imc/etc/IMCconfig
setenv PATH $INFORMIXDIR/bin:$PATH
```

For your convenience, these environment variable settings are also listed in two local files:
imc_0224.cshrc
imc_0224.kshrc

To activate the required environment variables, execute one of these files. For example, enter:

```
source imc.0224.cshrc.
```

For more information on the required environment variables, see [“Setting Environment Variables” on page 2-15](#).

Running the RUN_AS_ROOT.imc Script

You must run the **RUN_AS_ROOT.imc** script before you run MaxConnect. To run this script, make sure that you are logged in as user **root** and enter the name of the script, as the following example shows:

```
host2% su root
Password:
# ./RUN_AS_ROOT.imc
Informix Product:      Informix Connection Server
Installation Directory: /informix/imc
Performing root portion of installation of Informix Connection Server...
Installation of Informix Connection Server complete.
# exit
```

You are now ready to start MaxConnect. For instructions on how to start MaxConnect, see [“Starting MaxConnect” on page 4-7](#).

Administering MaxConnect

In This Chapter	4-3
How MaxConnect Works.	4-4
Listener Thread	4-5
Sessions	4-5
Worker Queues and Threads	4-5
Transports	4-6
Receive Threads	4-6
MaxConnect Poll Threads	4-6
Database Server Poll Threads	4-6
Starting and Stopping MaxConnect	4-7
Starting MaxConnect	4-7
Stopping MaxConnect	4-7
Using the Informix Server Administrator Utility.	4-8
Using ISA to Administer MaxConnect.	4-8
Using ISA to Monitor MaxConnect	4-8
Using the imcadmin Utility	4-9
Using the Command-Line Mode.	4-10
Using Interactive Mode	4-10
Syntax of the imcadmin Command	4-11
Monitoring MaxConnect on the Database Server Using the onstat Utility	4-25
Monitoring Error Messages	4-31

In This Chapter

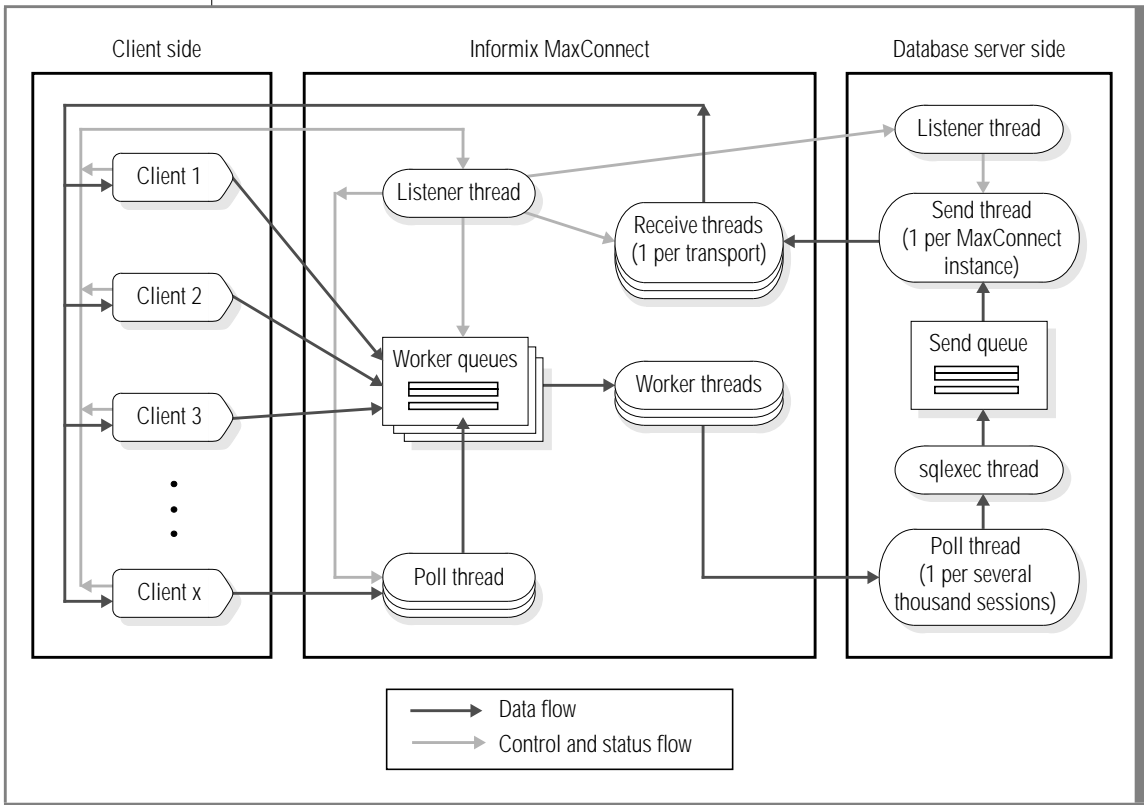
This chapter describes how to start, stop, monitor, and administer MaxConnect and contains the following sections:

- [How MaxConnect Works](#)
- [Starting and Stopping MaxConnect](#)
- [Using the Informix Server Administrator Utility](#)
- [Using the imcadmin Utility](#)
- [Monitoring MaxConnect on the Database Server Using the onstat Utility](#)
- [Monitoring Error Messages](#)

How MaxConnect Works

Client applications connect to MaxConnect exactly as they would to the database server. [Figure 4-1 on page 4-4](#) illustrates the interaction of the processes, or threads, within MaxConnect and between it and the application and database servers. The following sections describe these processes and how you can affect them.

Figure 4-1
Interprocess Communication



For more information on the interprocess communication of threads on the database server, refer to the [Administrator's Guide for Informix Dynamic Server 2000](#).

Listener Thread

The MaxConnect listener thread receives a connection request from a client application and establishes a connection to handle SQL requests from that client.

Sessions

A session begins when a client connects to MaxConnect.

The same session ID is used on both MaxConnect and on the database server. For example, the command `onstat -g ses 17` executed on the database server displays information about the same session that the command `imcadmin -session 17` on the MaxConnect server displays.

For information on how to specify the number of sessions that connect to the database server, see [“Updating the NETTYPE Configuration Parameter” on page 2-13](#). For information on how to monitor and performance tune, see [“Number of Sessions in NETTYPE” on page 5-8](#).

Worker Queues and Threads

Data packets from clients are placed in the data buffers of a worker queue, each of which is associated with a particular worker thread. The worker thread aggregates packets from different clients and sends the aggregated packets to the database server.

The IMCWORKERTHEADS configuration parameter determines the number of worker threads. The IMCWORKERDELAY configuration parameter determines the time that worker threads wait to accumulate packets before they perform an aggregated send.

For more information on these configuration parameters, see [“Specifying Configuration Parameters” on page 2-18](#). For information on tuning these configuration parameters for performance, see [“Number of Worker Threads” on page 5-16](#) and [“Worker Thread Delay and Packet Aggregation” on page 5-17](#). For information on worker-thread statistics, see [“The -worker Option” on page 4-17](#).

Transports

Each multiplexed connection between MaxConnect and the database server is called a transport. The number of transports is determined by the value of the IMCTRANSPTS configuration parameter. The installation script sets this value to the default of 2. For information on tuning this value, see [“Number of Transports” on page 5-11](#).

Receive Threads

The database server processes requests and sends packets back to the MaxConnect receive threads. These threads read the aggregated data, de-aggregate the data, and send the individual data packets to their corresponding clients. MaxConnect establishes one receive thread for each transport. The number of transports is specified by the IMCWORKERTRANSPORTS configuration parameter.

MaxConnect Poll Threads

MaxConnect automatically creates poll threads when it starts. Operations handled by the poll threads operate less efficiently than those that are handled directly by the worker queues. MaxConnect uses poll threads only when it encounters operating-system limits. MaxConnect does not encounter these limits if the kernel configuration is properly set. For information on kernel configuration, see the online machine notes referenced in [“Documentation Notes, Release Notes, Machine Notes” on page 10](#) of the Introduction. For information on monitoring poll threads, see [“The -poll Option” on page 4-24](#).

Database Server Poll Threads

The database server establishes one poll thread for each several thousand connections. With MaxConnect, a few transports can support thousands of client connections and you save system resources on the database server by reducing the number of poll threads. For more information, see [“Updating the NETTYPE Configuration Parameter” on page 2-13](#).

Starting and Stopping MaxConnect

To start or stop a MaxConnect process, you must be logged on as either user **root** or user **informix** on the computer where the process is running. You can start MaxConnect either before or after you start the database server.

Before you start MaxConnect:

- Complete the installation procedures outlined in [Chapter 3, “Installing MaxConnect,”](#) including editing the **sqlhosts** file and setting the MaxConnect configuration file.
- Set the **IMCSERVER**, **IMCADMIN**, **INFORMIXSERVER**, and **INFORMIXDIR** environment variables with appropriate values.

You do not need to set the **IMCCONFIG** and **INFORMIXSQLHOSTS** environment variables if this information is in the default location.

Starting MaxConnect

To start MaxConnect, use either of the following commands:

- `imcserver`

MaxConnect starts and prints its current configuration.

- `imcadmin -start`

The **imcadmin** utility locates the `imcserver` executable file **\$INFORMIXDIR/bin/imcserver** and starts `imcserver`. It also verifies that MaxConnect has established itself as a daemon process.

Stopping MaxConnect

Use the **imcadmin** utility to stop MaxConnect on the local computer. Enter the following command and reply `yes` to the confirmation prompt to stop MaxConnect:

```
> imcadmin -stop
Do you want to stop IMC? ('yes' to stop IMC) ==> yes
```

The **imcadmin** utility silently stops MaxConnect.

If you do not want to see the confirmation prompt, you can specify `yes` on the command line:

```
>imcadmin -stop yes
```

Using the Informix Server Administrator Utility

Informix Server Administrator (ISA) is a browser-based tool that provides Web-based system administration. You can use it to monitor and administer MaxConnect. In addition to using ISA to manage MaxConnect, you can use ISA for database server administration.

For information on how to download ISA, see the release notes file discussed in [“Documentation Notes, Release Notes, Machine Notes”](#) on page 10 of the Introduction. For information on using ISA, see online help.

Using ISA to Administer MaxConnect

With ISA, you can perform the following administrative tasks for MaxConnect:

- View the current settings for the MaxConnect configuration file and environment variables
- Change the MaxConnect configuration file
- View or change the `sqlhosts` file on the MaxConnect server
- Start or stop MaxConnect
- View the MaxConnect message log

Using ISA to Monitor MaxConnect

You can use ISA to monitor the effectiveness of the current values of MaxConnect configuration parameters. ISA uses information that `onstat -g imc` or `imcadmin` commands generate to display MaxConnect information, as the following table shows. Click the **Refresh** button to rerun the `onstat` or `imcadmin` command and display fresh information.

To monitor	Select on ISA	Displays the output of	Refer to
Number of sessions, as specified by the third field in the NETTYPE parameter	Database server that is processing client requests Performance→MaxConnect	onstat -g imc	“Monitoring and Tuning the Number of Sessions in NETTYPE” on page 5-10
Number of transports on database server	Database server that is processing client requests Performance→MaxConnect	onstat -g imc	“Monitoring Throughput of Transports and Tuning IMCTransports” on page 5-12
Number of transports on MaxConnect server	MaxConnect server Status→Transports	imcadmin -trans	“Using imcadmin -trans to Monitor and Tune IMCTransports” on page 5-13
Number of worker threads	MaxConnect server Status→Workers	imcadmin -worker	“Monitoring Worker Threads and Tuning IMCWORKERTHREADS” on page 5-16
Packet aggregation	MaxConnect server Status→Aggregation Status→Transports Status→Workers	imcadmin -aggr imcadmin -worker imcadmin -trans	“Monitoring Packet Aggregation” on page 5-17

Using the *imcadmin* Utility

You can use the **imcadmin** utility to start or stop MaxConnect on a local computer and to perform the following tasks on MaxConnect instances on local or remote computers:

- read the last lines of the log file
- monitor statistics on MaxConnect memory buffers, sessions, worker threads, transports, packet aggregation, packet size, status, and poll threads

You can use this utility from the command line or through an interactive mode. The **imcadmin** utility uses the TCP/IP communications protocol to connect to the MaxConnect process.

Any user can use the **imcadmin** utility to gather statistics. However to start or stop MaxConnect, you must be logged on as user **root** or user **informix** on the computer where MaxConnect resides.

For information on configuring the **imcadmin** utility, see [“Configuring the imcadmin Utility” on page 2-20](#).

Using the Command-Line Mode

To run the **imcadmin** utility from the command line, enter the full command including the options. For example, the following command displays statistics regarding CPU utilization and memory:

```
%imcadmin -cpu -mem
```

After you enter an option and the output is displayed, control returns to the command-line prompt.

Using Interactive Mode

To run the **imcadmin** utility in interactive mode, enter `imcadmin` with no arguments on the command line:

```
%imcadmin
```

The **imcadmin** utility displays the following prompt symbol in response:

```
==>
```

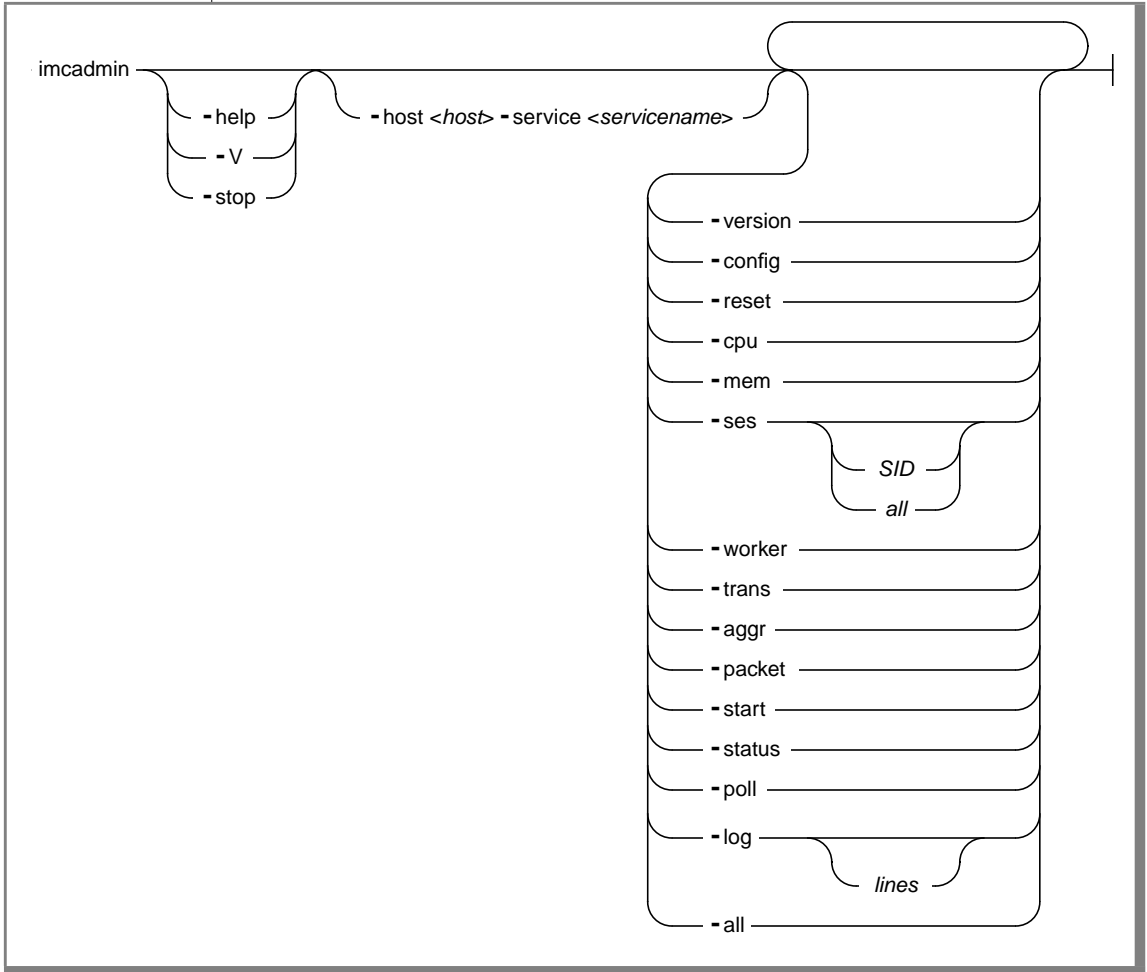
To see statistics regarding CPU utilization, enter `cpu` in response to this prompt:

```
==>cpu
```

The **quit** command allows you to exit the interactive mode of the **imcadmin** utility.

Syntax of the imcadmin Command

The following diagram shows the syntax of the **imcadmin** command.



Syntax of the `imcadmin` Command

Element	Purpose
<code>-help</code>	Prints a list of all of the options of the <code>imcadmin</code> command.
<code>-version</code>	Displays the version number and software serial number of the MaxConnect process.
<code>-V</code>	Displays the version number of the <code>imcadmin</code> utility.
<code>-config</code>	Prints the MaxConnect configuration file and environment variable settings.
<code>-reset</code>	Resets MaxConnect statistics.
<code>-cpu</code>	Displays CPU utilization by the MaxConnect processes since the last reset. Includes <code>usr_time</code> , the CPU time spent in user code and <code>sys_time</code> and the CPU time spent in operating system. Times are displayed both in seconds and as percentages.
<code>-mem</code>	Prints the memory statistics for the internal MaxConnect buffer manager. See “ The -mem Option ” on page 4-13.
<code>-ses</code>	Without any arguments, this option prints a summary of statistics for all sessions. See “ The -ses Option ” on page 4-14. With the session ID included as an argument, prints statistics for that session. See “ The -ses SID Option ” on page 4-16.
<code>-worker</code>	Displays statistics related to worker threads. See “ The -worker Option ” on page 4-17.
<code>-trans</code>	Displays statistics related to transports to the database server. See “ The -trans Option ” on page 4-18.
<code>-aggr</code>	Displays packet aggregation statistics for sends to the database server and receives from the database server. See “ The -aggr Option ” on page 4-21.
<code>-packet</code>	Displays distribution data about the size of network messages exchanged between MaxConnect and the database server and between MaxConnect and client applications. See “ The -packet Option ” on page 4-21.
<code>-start</code>	Starts MaxConnect. See “ Starting MaxConnect ” on page 4-7.
<code>-stop</code>	Stops MaxConnect. See “ Stopping MaxConnect ” on page 4-7.
<code>-status</code>	Gives a brief overview of some of the key statistics for a given MaxConnect instance.
<code>-poll</code>	Displays statistics related to the system calls that the poll threads use to scan for data that is to be received from connected clients.
<code>-log</code>	Prints the specified number of lines from the MaxConnect message-log file. If no line value is specified, prints the last ten lines.
<code>-all</code>	Includes outputs of all of the following options: <code>-version</code> , <code>-config</code> , <code>-cpu</code> , <code>-worker</code> , <code>-ses</code> , <code>-mem</code> , <code>-trans</code> , <code>-aggr</code> , <code>-packet</code> , <code>-poll</code> , and <code>-log</code> .
<code>-host <host></code>	Connects the <code>imcadmin</code> utility to the MaxConnect instance that the values of <code>host</code> and <code>servicename</code> specify. You can use this option from the command-line mode only, not from the interactive mode.
<code>-service <servicename></code>	

Additional information about each option that returns statistics is detailed in the following sections. These include statistics on MaxConnect memory buffers, sessions, worker threads, transports, packet aggregation, packet size, status, poll threads, and log files.

The -mem Option

The **imcadmin -mem** command prints the memory statistics for the MaxConnect buffer manager. Memory buffers of different sizes contain different free lists. The **-mem** option displays statistics for each of the free lists. The lists are sorted by size.

For more information on network buffer pools, see the [Performance Guide for Informix Dynamic Server 2000](#).

The following example shows sample output of the **-mem** option:

```
MEMORY_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,
reset at 10/07/99 16:39:29

36091 operations since reset, 40 ops/sec

list size #alloc #used #free allocKB
  1  540   300   300    0   158
  2   32    4    0    4    0
  3 4096   424   420    4  1696
Total -    728   720    8  1854
```

The following list describes each item in the output.

Output	Description
operations since reset	Number of accesses to the buffer manager module
ops/sec	Number of accesses per second to the memory manager
list	The index of the memory list
size	Size of the memory buffers maintained in the list This value cannot be accumulated for the different lists, so there is no value in the row labelled Total.
#alloc	Number of buffers allocated for each list

(1 of 2)

Output	Description
#used	Number of buffers in the list that are in use
#free	Number of buffers available in the free list
allocKB	Number of kilobytes allocated for the list

(2 of 2)

The -ses Option

The **imcadmin -ses** command without any arguments prints a summary of all session statistics. The following example shows sample output of the **-ses** option:

```
SESSION_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,
reset at 10/07/99 16:39:29
```

```
current      #phys  #act  #conn  #disc  #flow
            150   150   0      0      0

accumulated #total  #max  #mig  #spins #flow
            150   0     0     0      0

MAXof #KB_I2C  #pkt_I2C  #KB_C2I  #pkt_C2I
150   5316    16838   770    15527
AVGof #KB_I2C  #pkt_I2C  #KB_C2I  #pkt_C2I
150   2648    8601    329    7689
```

The following list describes each item in the output.

Output	Description
current	This section lists statistics on client sessions currently connected to MaxConnect.
#phys	Number of connected client sessions This value is the sum of the #act , #conn , and #disc columns.
#act	Number of fully established logical connections of clients to the database server
#conn	Number of pending sessions in the connect state
#disc	Number of sessions in the process of disconnecting

(1 of 2)

Output	Description
#flow	Number of sessions currently in a flow-controlled state That is, no packets will be sent from the client to the database server.
accumulated	This section lists session statistics for all sessions connected during the interval since the last statistics reset.
#total	Total number of sessions handled since the last reset
#max	Maximum number of concurrent sessions handled at any time since the statistics reset
#mig	Number of times that session-handling was passed from receive threads to poll threads or vice versa
#spins	Total number of times that all sessions had to spin to gain access to private memory buffers
#flow	Total number of times that sessions were placed in a flow-controlled state
MAXof	Maximum number of sessions that statistics are based on The section that starts with the word <i>MAXof</i> contains maximum values for the number of sessions displayed in the MAXof column.
AVGof	Average number of sessions that statistics are based on The section that starts with the word <i>AVGof</i> contains average values for the number of sessions displayed in the AVGof column.
#KB_I2C	Number of kilobytes of data sent from MaxConnect to clients
#pkt_I2C	Number of packets sent from MaxConnect to clients
#KB_C2I	Number of kilobytes of data sent from clients to MaxConnect
#pkt_C2I	Number of packets sent from clients to MaxConnect

(2 of 2)

The -ses SID Option

The **imcadmin -ses SID** command displays statistics for the session specified in *SID*. The MaxConnect session ID is the same as the one assigned by the database server. If you use the **-ses all** option in place of **-ses SID** statistics for all sessions, the **imcadmin** utility displays statistics for each session.

The following example shows sample output of the **-ses all** option:

```
SESSION_stats for 224 seconds at 11/29/99 15:28:17 up for 00:03:44,
reset at 11/29/99 15:24:33

ses_id state #KB_C2I #PCK_C2I #KB_I2C #PCK_I2C #mig #flow #spins mode
  23 Act      3         62  42165  10586  0    0    0 recv
  28 Disc     1         22   2068    483  0    0    0 recv
```

The following list describes each item in the output.

Output	Description
ses_id	Session ID of the session.
state	Current session state, possible values are: <ul style="list-style-type: none"> ■ Act: active database session ■ Conn: connection to database server is being established. ■ Disc: session is disconnecting. ■ Flow: session is flow controlled.
#KB_C2I	Number of kilobytes of data that the client sent to MaxConnect
#PCK_C2I	Number of packets sent by client to MaxConnect
#KB_I2C	Number of kilobytes of data that MaxConnect sent to the client
#PCK_I2C	Number of packets that MaxConnect sent to the client
#mig	Number of times a session changed its mode
#flow	Number of times a session was placed in a flow state

(1 of 2)



Output	Description
#spins	Number of times the session receive thread was spinning because of a shortage of network buffers
mode	Mode of a session, possible values for this field are: <ul style="list-style-type: none"> ■ read: indicates a dedicated thread to perform receives for the session. ■ poll: indicates that the session was handled by means of a poll thread. In general, this value indicates a setup problem.

(2 of 2)

Tip: C2I means from client to Informix MaxConnect. I2C means from Informix MaxConnect to client.

The -worker Option

The **imcadmin -worker** command displays statistics related to worker threads and a histogram of arrival rates in intervals of 50 microseconds up to 1000.

The first column displays the sum of the subsequent columns that display the statistics for each interval. For example, the last column shows how often the worker thread was wakened more than 1000 microseconds after it finished its previous activity cycle. The worker thread was wakened 237,053 times with an inactivity period of larger than 1000 microseconds. In the example this represents 53.8 percent of the samples.

```

WORKER_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,
reset at 10/07/99 16:39:29

workers      wakeups      sleep(s)      avg(us)      delayed
  1          440465      869.764236      1739          0

#dequeues   MaxQlen   AvgQlen      Sends   MaxSpins   AvgSpins
  461444         8     1.02      455834         0     0.00

arrival interval histogram (micro seconds)
sum <50 <100 <150 <200 <300 <350 <400 <500 <750 <1000 >=1000
440465 2703 15000 15549 14204 25775 2062 12065 22301 6761 36992 237053

100.0% 0.6   3.4   3.5   3.2   5.9   2.7   2.7   5.1  10.6   8.4   53.8

```

The following list describes each item in the first section of the output.

Output	Description
workers	Number of worker threads evaluated
wakeups	Number of times the worker thread was activated
sleep(s)	The time the worker thread was inactive
avg(us)	Average duration in microseconds of intervals between wakeups
delayed	Number of packets that were delayed as a result of the current value of the IMCWORKERDELAY parameter
#dequeues	Number of times a worker queue was emptied
MaxQlen	Maximum length of queues the worker thread accessed.
AvgQlen	Average length of a worker queue
sends	Number of times that the worker thread sent an aggregated packet to the database server
MaxSpins	Maximum number of spins the worker had to perform in order to wait for a free transport
AvgSpins	Average number of spins that the worker had to perform. This value is always 0 when only one worker thread is configured

The -trans Option

The **imcadmin -trans** command displays statistics related to transports to the database server. The following example shows sample output of the **-trans** option:

```
TRANSPORT_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,  
reset at 10/07/99 16:39:29
```

```
Write statistics for 7 transports:
```

```
writes      total  avg  max  avg  max  partial  KB writes  
            KB  bytes bytes aggr aggr  writes  /sec  /sec  
455835     170453  382  4700  1.0  8      0 189.39  506
```

```

Read statistics for 7 transports:
  reads      total  avg  max  avg  max  partial  KB  reads
            KB  bytes bytes aggr aggr  reads  /sec /sec
406778      397227  999 26664  1.3  15   38674  441.36  451

#transport  #out of
  spins    sequence
    0      9023

```

The statistics section named `Send statistics for X transports` includes the following output fields (where `X` is the number of transport connections established between MaxConnect and the database server).

Output	Description
writes	Number of send operations from MaxConnect to the database server
total KB	Number of kilobytes sent on all transports
avg bytes	Average number of bytes per send operation
max bytes	Maximum number of bytes sent per send operation
avg aggr	Average aggregation, the average number of client-side network messages that have been combined into one larger network message
max aggr	Maximum aggregation, the maximum number of client-side network messages that have been combined into one larger network message
partial writes	Number of sends that the operating system fragmented
KB/second	Average number of kilobytes of data that were transferred per second
writes/sec	Average number of send operations per second

The statistical section that begins `Recv` statistics for `X` transports displays the following values. `X` refers to the number of transport connections established between `MaxConnect` and the database server.

Output	Description
reads	Number of receive operations for data sent from the database server to <code>MaxConnect</code>
total KB	Number of kilobytes received on all transports
avg bytes	Average number of bytes received per receive operation
max bytes	Maximum number of bytes received with any receive operation
avg aggr	Average aggregation, the average number of network messages received from the database server that have been combined into one larger network message
max aggr	Maximum aggregation, the maximum number of client-side network messages that the database server combined into one larger network message.
partial reads	Number of receive operations that the operating system partitioned
KB/sec	Average number of kilobytes of data that were transferred per second
reads/sec	Average number of receive operations per second

Two other statistical values are displayed near the bottom of the output.

Output	Description
#transport spins	Number of times a write operation had to perform a long spin to get a free transport
# out of sequence	Number of packets that had to be queued for later delivery

The -aggr Option

The **imcadmin -aggr** command displays packet aggregation statistics for sends to the database server and receives from the database server. The following example shows sample output of the **-aggr** option:

```
AGGREGATION_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,
reset at 10/07/99 16:39:29

IMC to database aggregation histogram [aggregation count]
  sum      <2    <4    <6    <8    <10   <12   <14   >=14
455835  441189  14627    16     2     1     0     0     0
100.0%   96.8    3.2    0.0    0.0   0.0   0.0   0.0   0.0

database to IMC aggregation histogram [aggregation count]
  sum      <2    <4    <6    <8    <10   <12   <14   >=14
406772  337198  56346  10321  2581   232    44    22    28
100.0%   82.9   13.9    2.5    0.6   0.1   0.0   0.0   0.0
```

The first histogram describes the aggregation distribution for data sent from clients to the database server. The second histogram shows aggregation distribution for data sent from the database server to MaxConnect and to client applications. The histograms display the number of network messages received in each range.

The -packet Option

The **imcadmin -packet** command displays distribution data about the size of network messages exchanged between MaxConnect and the database server and between MaxConnect and client applications. The **imcadmin** utility displays the size of packets, in bytes, for packets transferred since the last reset in these categories.

- Packets received from clients
- Packets sent to clients
- Packets received from the database server
- Packets sent to the database server

The following example shows sample output of the **imcadmin -packet** command. The first output row in each category represents count and the second represents percentages.

```
PACKET_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,
reset at 10/07/99 16:39:29
```

```
Packet size client to IMC
  sum      <256  <512  <1024  <1540  <2048  <4096  >=4096
470939  431191  18198  15187   4114     0    2249     0
100.0%   91.6    3.9    3.2    0.9    0.0    0.5    0.0
```

```
Packet size IMC to client
  sum      <256  <512  <1024  <1540  <2048  <4096  >=4096
528287  316165  83672  47013  2953    596   20548  57340
100.0%   59.8   15.8    8.9    0.6    0.1    3.9   10.9
```

```
Packet size database to IMC
  sum      <512  <1024  <1540  <2048  <3072  <4096  >=4096
406778  219833  79959  30506  2944   2978   4469   66089
100.0%   54.0   19.7    7.5    0.7    0.7    1.1   16.2
```

```
Packet size IMC to database
  sum      <512  <1024  <1540  <2048  <3072  <4096  >=4096
455835  415970  22798  12153  2648   2259     5     2
100.0%   91.3    5.0    2.7    0.6    0.5    0.0    0.0
```

The **sum** column in each histogram displays how many packets were evaluated in total. As an example of how to interpret the histograms, consider the column labelled **<1024** in the histogram titled **Packet size IMC to database**. The number **22,798** in this column means that **22,798** out of **455,835** packets sent from MaxConnect to the database server ranged in size from 512 bytes to 1023 bytes, which represents 5.0 percent.

The -status Option

The **imcadmin -status** command gives a brief overview of some of the key MaxConnect statistics.

```
STATUS_stats for 60 seconds at 12/02/99 17:45:19 up for 00:01:00,
reset at 12/02/99 17:44:19
```

```
threads  sessions  clt_reads  clt_writes  selects  dry
      9      0         0         0         0         0

trans    spins    srv_reads  srv_writes
      0      0         0         0

      mem    usr_cpu  sys_cpu  avg_qlen
      60    0.05    0.02    0.00
```

The following table describes each item in the output.

Output	Description
threads	Number of threads currently active for MaxConnect
sessions	Number of client sessions connected to MaxConnect
clt_reads	Number of receive operations from client sessions since the last reset
clt_writes	Number of send operations to clients since the last reset
selects	Number of poll operations executed since the last reset
dry	Number of poll operations without events since the last reset
trans	Number of transports to the database server that are currently established
spins	Number of spins until transports became available, relative to the last reset
srv_reads	Number of receive operations from the database server since the last reset
srv_writes	Number of send operations to the database server since the last reset
mem	Number of kilobytes the MaxConnect buffer manager allocates

(1 of 2)

Output	Description
usr_cpu	CPU seconds of user time since the last reset
sys_cpu	CPU seconds of system time since the last reset
avg_qlen	Average length of worker queues since the last reset

(2 of 2)

The -poll Option

The **imcadmin -poll** command displays statistics related to the system calls that are used by the MaxConnect poll threads to scan for data sent by connected clients. The following example shows the output of the **-poll** option:

```
POLL_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,
reset at 10/07/99 16:39:29
```

```

#poll    #fds    #poll    #dry    avg#fds    #poll
threads /thread  ops      polls   /poll     ops/sec
      8     256      0       0         -         0.00
```

The following table explains the meaning of each item in the output.

Output	Description
#poll threads	Number of threads that MaxConnect creates to handle poll operations for all connected clients
#fds thread	Maximum number of client-side connections that each poll thread handles
#poll ops	Total number of poll operations
#dry polls	Number of poll operations that did not return any event data
avg#fds/poll	Average number of client-side events detected per poll operation
#poll ops/sec	Average number of poll operations executed per second during the interval stated in the header line. In this example the interval is 900 seconds

The -log Option

The **imcadmin -log [*lines*]** command prints a specified number of lines from the end of the MaxConnect message-log file. If you omit the *lines* parameter, the **-log** option prints the last 10 lines of the log file. The following example shows a sample output of the **-log** option:

```

info(11:09:18): RESET at 03/27/00 11:09:18: - initialization
info(11:09:18): imcserver: creating 4 poll threads to support (approx) 1013
sessions
info(11:09:18): imcserver: started listener and pollthreads
info(11:09:23): imcserver: opened 3 transports to DBS
info(11:09:26): imcserver: Admin thread has requested shutdown
info(11:09:26): imcserver: done and exiting now

info(11:09:33): RESET at 03/27/00 11:09:33: - initialization
info(11:09:33):imcserver: creating 4 poll threads to support (approx) 1013
sessions
info(11:09:33): imcserver: started listener and pollthreads

```

Monitoring MaxConnect on the Database Server Using the onstat Utility

You can monitor MaxConnect operations on the database server with certain options of the **onstat** utility, which the following table describes.

onstat -g Option	Topic or Function
onstat -g imc	Prints information about MaxConnect instances that are connected to the database server. For more information, see “The onstat -g imc Command” on page 4-26 .
onstat -g ntd	Prints network statistics by service.
onstat -g ntm	Prints network mail statistics.

(1 of 2)

<code>onstat -g</code> Option	Topic or Function
<code>onstat -g ntt</code>	Prints network user times. Contains MaxConnect specific information described on page 4-30 .
<code>onstat -g ntu</code>	Prints network user statistics. Contains MaxConnect specific information described on page 4-30 .
<code>onstat -g nta</code>	Prints combined network statistics from <code>onstat -g imc</code> , <code>onstat -g ntd</code> , <code>onstat -g ntm</code> , <code>onstat -g ntt</code> , and <code>onstat -g ntu</code> . For more information on any of these options, see the Informix Administrator's Reference .

(2 of 2)

The `onstat -g imc` Command

The `onstat -g imc` command displays the MaxConnect statistics on the database server. In contrast, the `imcadmin` utility displays the MaxConnect statistics on the MaxConnect server.

The following example shows sample output for `onstat -g imc`:

```
Informix MaxConnect Instances
global network information:
#netscb connects   read   write   q-free  q-limits  q-exceed alloc/max
8/   9       3     106    107     2/   2  135/ 10    0/   0   7/   7

IMCid transports admport hostname
1           3     27015  porter.pdx.informix.com

10 statistics for each MaxConnect instance:
IMCid  header  data  partial  blocked  data  partial  blocked
      reads  reads  reads   reads   reads  writes  writes  writes
      1     112   109     0       0    107     0       0

Network Select/Poll system call data:
IMCTliPoll#: 0 selects: 1328953 empty: 1328839 fds/select: 0.0 (max: 2)

histogram [fds / select]
filedescs  selects  percent
0 - 1      113     99.12
2 - 3         1     0.88
sum         114    100.00
```

The first portion of the `onstat -g imc` output lists global network information that does not contain specific MaxConnect statistics. The following `onstat` options display this same global network information: `onstat -g imc`, `onstat -g ntd`, `onstat -g ntm`, `onstat -g ntt`, `onstat -g ntu`.

The global network information portion of these `onstat` options consists of the following columns.

Column	Description
#netscb	Current and maximum number of network session control blocks allocated since initialization or reset of statistics (<code>onstat -z</code> or <code>imcadmin -reset</code>)
connects	Total number of connection requests since initialization
read	Number of network reads performed by all threads
write	Number of network writes performed by all threads
q-free	Current and maximum number of free network buffers since initialization
q-limits	Threshold limit for buffers on the free and normal network buffer queues For information on how the <code>NETTYPE</code> configuration parameter affects this value, see “Network Free Buffer Pool” on page 5-9 .
q-exceed	Number of times the threshold for buffers on the free and normal network buffer queues has been exceeded
alloc/max	Number of network buffers currently allocated and the maximum number of network buffers allocated since initialization or reset of statistics

The second portion of the **onstat -g imc** option lists the following information for each MaxConnect instance connected to the database server.

Column	Description
IMCid	Internal identification number of each MaxConnect instance connected to the database server
transports	Number of actual network connections actually established between MaxConnect and the database server This value is the current value of the IMCWORKERTRANSPORTS configuration parameter.
admport	Port number on which the MaxConnect instance listens for connections from the imcadmin utility This value is the port number specified in the sqlhosts entry for the imcadmin utility.
hostname	Name of the host node on which the MaxConnect server is running This value is the hostname specified in the sqlhosts entry for MaxConnect.

The third portion of the **onstat -g imc** option lists the following I/O statistics.

Column	Description
header reads	Number of packet headers read. The packet header describes the structure of the data that follows.
data reads	Number of packets from which data has been read
partial reads	Number of reads that were too large to be completed in a single operation
blocked reads	Number of requested reads that could not be performed because of an unavailable resource. If this value is much greater than zero, there are too few transports or the network bandwidth is too low.

(1 of 2)

Column	Description
data writes	Number of packets to which data has been written
partial writes	Number of writes that were partitioned by the operating system because they are too large to be completed in a single operation
blocked writes	Number of requested writes that could not be performed because there are too few transports or network bandwidth is too low

(2 of 2)

The fourth portion section of the **onstat -g imc** output contains the following network select and poll system call data.

Column	Description
IMCTliPoll# or IMCSocPoll#	Internal ID of the poll thread
Selects	Number of poll or select calls executed
Empty	Number of poll or select calls that did not find any network data to process
fds/select	Average number of file descriptors with pending network data for each select or poll call

The last section of the **onstat -g imc** output contains a histogram with the following data.

Column	Description
filedescs	Number range of file descriptors
Selects	Number of select calls that used the range of file descriptors
Percent	Percent of all selects for each file descriptor range

The onstat -g ntt Command

The **onstat -g ntt** and the **onstat -g ntu** options display a line for each transport established between a MaxConnect instance and the database server. The database server does not associate a MaxConnect transport with a session or thread. Each transport line has the value `imctransX` in the **Thread Name** column, where `X` is a number that identifies a particular MaxConnect instance.

The onstat -g ntu Command

The **onstat -g ntu** option displays the following threads or network session control block for MaxConnect on the database server.

Thread Name	Description
tliimclstnr or socimclstnr	The listener thread that accepts MaxConnect network connections
tliimcpoll or socimcpoll	The poll thread that accepts handle communication over MaxConnect network connections
imcsndX	The dedicated send thread that services send requests to each MaxConnect instance The value of <code>X</code> is the unique integer number that the database server assigns to each MaxConnect instance that is connected to it.

Monitoring Error Messages

MaxConnect passes error messages from the database server to the client. For example, if MaxConnect cannot connect to the database server, the client generates the same error code that it would generate if the database server were down and the user receives the same response as if the client tried to connect to a database server instance that is offline.

After you start the MaxConnect process, MaxConnect-specific errors are logged to the MaxConnect log file. All MaxConnect error messages are unnumbered. Each message is identified by type using one of the following tags:

- **info**

These messages provide information on MaxConnect status or operation.

- **warn**

These messages provide warnings.

- **fatal**

These messages indicates serious problems. For example, messages that indicate that MaxConnect is out of resources or that an internal error occurred such as memory corruption.

For an example of the messages printed to the MaxConnect log file, see [“The -log Option” on page 4-25](#).

Tuning Performance

In This Chapter	5-3
Planning and Configuring for Performance	5-3
Number of Poll Threads	5-4
Number of Network Protocols	5-4
Process Priority of MaxConnect	5-6
Multiplexed Connections Using NETTYPE sqlmux and MaxConnect Together	5-6
Use of MaxConnect with Wide-Area Networks	5-7
WAN Links and Network Routers	5-7
WAN Links and Number of MaxConnect Transports	5-8
Monitoring and Tuning Configuration Parameters	5-8
Number of Sessions in NETTYPE	5-8
Network Free Buffer Pool	5-9
Monitoring and Tuning the Number of Sessions in NETTYPE.	5-10
Number of Transports	5-11
Monitoring Throughput of Transports and Tuning IMCTRANSPTS	5-12
Using imcadmin -trans to Monitor and Tune IMCTRANSPTS	5-13
Changing the Number of Transports	5-15
Number of Worker Threads	5-16
Monitoring Worker Threads and Tuning IMCWORKERTHEADS	5-16
Changing the Number of Worker Threads	5-16
Worker Thread Delay and Packet Aggregation	5-17
Monitoring Packet Aggregation.	5-17
Tuning IMCWORKERDELAY	5-21

Load Balancing Across Multiple Instances	5-23
Establishing Server Groups on UNIX	5-23
Establishing Server Groups on Windows NT.	5-24
Providing for MaxConnect Failure	5-25

In This Chapter

This chapter provides techniques and scenarios for improving and controlling performance in the MaxConnect environment. The topics are divided into the following categories:

- [Planning and Configuring for Performance](#)
- [Monitoring and Tuning Configuration Parameters](#)
- [Load Balancing Across Multiple Instances](#)
- [Providing for MaxConnect Failure](#)

Planning and Configuring for Performance

Many decisions that you make while planning the configuration of MaxConnect can affect the performance of the system or user response times. Chapter 2 describes the details for the following decisions:

- Multiple MaxConnect instances
- Remote versus local MaxConnect instance relative to the client or application server
- Network protocol choice
- Use of old protocols (TLITCP or SOCTCP)
- Number of poll threads and virtual processor choices in NETTYPE parameter
- Configuring for applications that access multiple databases

The following sections describe these additional planning considerations:

- [Number of Poll Threads](#)
- [Number of Network Protocols](#)
- [Process Priority of MaxConnect](#)
- [Multiplexed Connections Using NETTYPE sqlmux and MaxConnect Together](#)
- [Use of MaxConnect with Wide-Area Networks](#)

Number of Poll Threads

In previous versions of the database server, you might have configured multiple poll threads. With MaxConnect you can reduce the number of poll threads to one or two. A smaller number of poll threads reduces the CPU overhead of managing multiple poll threads on the database server.

Number of Network Protocols

When you first configure MaxConnect, it is easier to define a new network protocol entry for the MaxConnect instance in the client **sqlhosts** file, as [“Modifying the sqlhosts File” on page 2-16](#) describes. However, each additional entry creates one additional listener thread and associated poll threads, which might increase CPU utilization on the database server.

Because MaxConnect supports the old protocols (TLITCP or SOCTCP) as well as TLIIMC and SOCIMC, you can reduce the number of listener threads and poll threads to one each. A smaller number of listener and poll threads reduces the CPU overhead on the database server.

To reduce the number of listener threads

1. Create another **sqlhosts** file on the database server that specifies the MaxConnect protocol. The following table shows two different **sqlhosts** files.

Pathname	dbservername	nettype	hostname	servicename
\$INFORMIXDIR/ etc/sqlhosts_tcp	miami	ontlitcp	host1	portD
\$INFORMIXDIR/ etc/sqlhosts_imc	miami	ontliimc	host1	portD

2. Change the **INFORMIXSQLHOSTS** environment variable to specify the pathname of the **sqlhosts** file that specifies the MaxConnect protocol and start the database server.


```
setenv INFORMIXSQLHOSTS $INFORMIXDIR/etc/sqlhosts_imc
```
3. To allow execution of command-line utilities such as **oncheck** and **dbaccess**, change the **INFORMIXSQLHOSTS** environment variable to specify the pathname of the **sqlhosts** file that specifies the old protocol.

```
setenv INFORMIXSQLHOSTS $INFORMIXDIR/etc/sqlhosts_tcp
```

You do not need to change the **INFORMIXSERVER** environment variable when you use this method of changing **sqlhosts** files.

4. Delete the **tlitcp** or **soctcp** **NETTYPE** entry from the **ONCONFIG** file.
5. Change the value of **DBSERVERNAME** back to the original setting (**miami** in the example) in the **ONCONFIG** file.
6. Delete from the **DBSERVERALIASES** list the entry for **miami** (because it is now the **DBSERVERNAME**).
7. Set **INFORMIXSERVER** environment variable to its original setting (**miami** in the example).
8. Restart the database server.



Process Priority of MaxConnect

MaxConnect executes with a fixed, low real-time priority on most platforms. On hardware platforms that do not have real-time priorities, MaxConnect executes with a high timeshare priority.

Important: Do not change the priority with which MaxConnect is running. Otherwise, user response times and throughput might degrade.

Multiplexed Connections Using NETTYPE sqlmux and MaxConnect Together

If you have a large number of nonthreaded client applications that open many connections to the database server, you can use the multiplexed connection facility (NETTYPE sqlmux) and MaxConnect together to achieve two levels of multiplexing.

- The NETTYPE sqlmux facility multiplexes all the connections in a single nonthreaded application to one network connection.
- MaxConnect multiplexes many applications to one connection.

These multiplexed connections provide the following benefits for nonthreaded ESQL/C clients:

- Decreased number of network connections
- Improved response times by decreasing system CPU time

MaxConnect can provide the following additional performance benefits for existing nonthreaded ESQL/C applications that use the multiplexed connection facility:

- Decreased physical connections across multiple clients, including multi threaded clients
- Improved response times because it uses packet aggregation, which reduces CPU utilization on the database server

You can use existing multiplexed connections with MaxConnect by changing the client **sqlhosts** entry, as the following table shows.

	dbservername	nettype	hostname	servicename	option
Original entry	miami	ontlitcp	host1	portA	m=1
Updated entry	miami	ontlitcp	host2	portB	m=1

Use of MaxConnect with Wide-Area Networks

You can configure MaxConnect in an existing Wide-Area Network (WAN) environment in the following ways:

- Between MaxConnect and the database server
- Between the clients and MaxConnect

When the WAN link is between the clients and MaxConnect, you do not need to take any special actions.

When the WAN link is between the database server and MaxConnect, you need to pay attention to the performance considerations that the following sections describe.

WAN Links and Network Routers

Your configuration might include network routers to route all TCP/IP network requests from a specific IP source address to a specific network link.

When you introduce MaxConnect in such a configuration, the router can force all MaxConnect network traffic through a single network link. This single network link can become a bottleneck that can degrade database throughput or user response times.

To help relieve this bottleneck, take one of the following actions:

- Obtain a new IP address. You obtain a new IP address by adding a network card.
- Use multiple dedicated MaxConnect computers.

WAN Links and Number of MaxConnect Transports

A WAN link (such as a T1 line) can be substantially slower than a LAN link (such as FDDI). If you notice a degradation in database throughput or user response times when using a WAN link, use the **onstat -g imc** utility and look at the **blocked writes** output column. If the value in the **blocked writes** output column is nonzero for the MaxConnect instance, you might improve performance if you configure more transports in the IMCTransports parameter.

Monitoring and Tuning Configuration Parameters

You can monitor the effectiveness of the current values of the number of sessions in the NETTYPE configuration parameter on the database server.

You can also monitor the effectiveness of the current values of the following configuration parameters on MaxConnect:

- Number of transports in IMCTransports
- Number of worker threads in IMCWorkerThreads
- Worker thread delay (IMCWorkerDelay) and packet aggregation

Number of Sessions in NETTYPE

With MaxConnect, the number of poll threads on the database server does not influence the total number of sessions. You must specify the total number of sessions connecting to the database server by modifying the third field of the NETTYPE configuration parameter in the ONCONFIG file.

Although the database server uses a default value for the number of sessions or can dynamically determine the number of sessions, you must specify the number of sessions when using MaxConnect for the following reasons:

- The database server uses the number of sessions specified in the NETTYPE configuration parameter to:
 - Determine the number of network buffers to maintain in the network free buffer pool.
 - Determine internal data structures sizes.
- The MaxConnect installation script uses the number of sessions to determine the number of transports to configure.

Network Free Buffer Pool

The database server dynamically allocates network buffers from the global memory pool for request messages from clients. After the database server processes client requests, it returns buffers to a common network buffer pool that is shared among sessions that use SOCTCP, IPCSTR, TLITCP, TLIIMC, or SOCIMC network connections.

The free network buffer pool can grow during peak activity periods. To prevent large amounts of unused memory from remaining in these network buffer pools when network activity is no longer high, the database server returns free buffers when the number of free buffers reaches specific thresholds. If the number of free buffers is greater than the threshold, the database server returns the memory allocated to buffers over the threshold to the global pool.

This threshold correlates the number of free network buffers with the number of sessions that you specify in the NETTYPE configuration parameter.

The database server uses the following formula to calculate the threshold for the free buffers in the network buffer pool:

$$\text{free network buffers threshold} = 100 + (0.7 * \text{number_sessions})$$

For more information on tuning network buffers, refer to your [*Performance Guide for Informix Dynamic Server 2000*](#).

Monitoring and Tuning the Number of Sessions in NETTYPE

To check if the number of sessions specified in the third field of the NETTYPE configuration parameter provides enough network buffers for the current workload, use any of the following methods:

- **onstat -g imc** command or the **Performance→MaxConnect** option on ISA on the database server

The output is the same for both ISA and **onstat -g imc**, which shows the network buffer statistics from the database server view point.
- **imcadmin -ses** command or the **Status→Sessions** option on ISA on the MaxConnect server

The output is the same for both ISA **Status→Sessions** and **imcadmin -trans**, which shows session statistics on MaxConnect.

The following example shows the `global network information` portion in sample output for **onstat -g imc** and ISA **Performance→MaxConnect**:

```
Informix MaxConnect Instances

global network information:
#netscb connects    read    write    q-free  q-limits  q-exceed alloc/max
72/ 73      85    2646    2750    20/21   114/ 10    0/ 0   105/105
```

To monitor the number of session with **onstat -g imc**, look at the following columns in the `global network information` portion of the output.

Output Column	Description
q-limits	In the previous example, the q-limits column shows a value of 114 for the threshold limit for the shared network free buffer pool, which the database server derives with the formula described in “Network Free Buffer Pool” on page 5-9.
q-exceed	The number of times that the threshold for the shared network free-buffer pool was exceeded. In most cases, the value in this column should be 0, as in the previous example.
alloc/max	The number of currently allocated network buffers and the maximum number of network buffers since the database server was initialized or statistics were reset (by either onstat -z or imcadmin -reset). The example above shows 105 currently allocated network buffers and 105 maximum allocated network buffers.

If the value in the **q-exceed** column is much greater than 0, the database server is spending a lot of processing time allocating and deallocating network buffers. One of the following problems exist:

- Not enough buffers are allocated for the number of sessions currently executing

Compare the value in the **alloc/max** column with the number of sessions specified in the third field of the **NETTYPE** configuration parameter. If the **alloc/max** value is much larger, increase the number of sessions specified in the **NETTYPE** configuration parameter.

- Not enough private buffers are allocated for each session

If the **alloc/max** value is less than or equal to the number of sessions specified in the **NETTYPE** configuration parameter, each session might have too few buffers for the amount of data transferred. Increase the number of private network buffers for each session with the **IFX_NETBUF_PVPOOL_SIZE** environment variable.

For more information on the **IFX_NETBUF_PVPOOL_SIZE** environment variable, refer to the [Informix Guide to SQL: Reference](#).

Number of Transports

To monitor statistics for MaxConnect transports, you can use any of the following methods:

- **onstat -g imc** command or the **Performance→MaxConnect** option on ISA option on ISA on the database server

The output is the same for both ISA and **onstat -g imc**, which shows the transport statistics from the database server view point.

- **imcadmin -trans** command or the **Status→Transports** option on ISA on the MaxConnect server

The output is the same for both ISA and **imcadmin -trans**, which shows transport statistics on MaxConnect.

Monitoring Throughput of Transports and Tuning IMCTransports

To monitor the throughput of transports with `onstat -g imc`, look at the information for each MaxConnect instance and the I/O statistics portions of the output. The following excerpt shows sample output for the `onstat -g imc` and `ISA Status→Transports` option:

```
IMCid transports admport hostname
  1           3 27015 porter.pdx.informix.com

I/O statistics for each MaxConnect instance:
IMCid  header  data  partial  blocked  data  partial  blocked
      reads  reads  reads   reads   reads  writes  writes  writes
  1    112   109     0     0    107     0     0
```

The **transports** column shows the current value of the IMCTransports configuration parameter. In the example, the **transports** column shows that 3 transports are currently configured.

The I/O statistics for each MaxConnect instance portion of the `onstat -g imc` output shows actual throughput values for the transports. The above example shows the following throughput values:

- The **data reads** column shows that 109 data packets have been read.
- The **data writes** column shows that 107 data packets have been written.
- The **blocked reads** column shows that 0 requested reads could not be performed.

You might choose to modify your configuration according to values you see in the following columns:

- The **blocked reads** column shows the number of requested reads that could not be performed from MaxConnect to the database server. If the value in the **blocked reads** column is much greater than 0, the network bandwidth might be too low.

Consider one of the following actions:

- Upgrading to a network card that has a larger maximum transfer unit (MTU) size.
- Changing the framing size of the network card.

- The **blocked writes** column shows the number of requested writes that could not be performed from MaxConnect to the database server because of an unavailable resource. If the value in the **blocked writes** column is much greater than zero, the number of transports might be too low.

For information on how to increase the value in the IMCTransports configuration parameter, refer to [“Changing the Number of Transports” on page 5-15](#).

Using imcadmin -trans to Monitor and Tune IMCTransports

The following sample output shows an excerpt for **imcadmin -trans** and **ISA Status→Transports** option:

```
TRANSPORT_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,
reset at 10/07/99 16:39:29
```

```
...
```

```
Read statistics for 4 transports:
```

reads	total	avg	max	avg	max	partial	KB	reads
	KB	bytes	bytes	aggr	aggr	reads	/sec	/sec
406778	397227	999	26664	1.3	15	38674	441.36	451

#transport	#out of
spins	sequence
0	9023

Subsequent sections describe how to interpret the following columns in the different portions of the **imcadmin -trans** output:

- **partial reads, reads, KB/second** and **avg aggr** columns in the **Read Statistics** portion
- **#transport spins** and **#out of sequence** columns at the bottom

Partial Reads, Reads, and Average Aggregation

The **partial reads** column in the **imcadmin -trans** and the **onstat -g imc** outputs displays the number of reads that MaxConnect could not receive in one transfer operation. Partial reads occur when TCP/IP splits a large network message into smaller transfer units.

Partial reads might affect user response times and network throughput because they require additional processing to receive multiple messages.

You might have network problems if the value in the **partial reads** column is greater than 20 percent of the product of **actual reads** and the average aggregation (**avg aggr** column), as the following formula shows:

```
partial reads > ((actual reads) * (avg aggr) * .2)
```

The values in the preceding sample **imcadmin -trans** output indicate that there is no network problem because the **partial reads** value (38674) is not greater than 20 percent:

```
partial reads > (.2 * (actual reads) * (avg aggr))
                (.2 * (406778) * 1.3))
                (.2 * (528811.4))
                105762.28
38674 < 105762.28
```

If the **partial reads** value is greater than 20 percent, consider one of the following actions:

- Upgrading to a network card that has a larger MTU size.
- Changing the framing size of the network card.

Transport Spins

The **#transport spins** column in the **imcadmin -trans** output shows the number of times MaxConnect had to wait for a transport to be free. Normally, the value in this column is 0, as the preceding sample output shows.

If the value in the **#transport spins** column is much greater than zero, the number of transports might be too low. For information on how to increase the value in the IMCTransports configuration parameter, refer to [“Changing the Number of Transports” on page 5-15](#).

Out-of-Sequence Packets and Kilobytes Per Second

Out-of-sequence packets might affect user response times and MaxConnect CPU utilization because they require additional time to send the packets in sequence to the client. The database server splits messages larger than 4 kilobytes into multiple packets less than or equal to 4 kilobytes. When the database server sends two packets on behalf of a single session at nearly the same time on different transports, MaxConnect might receive the second packet first. In this case, MaxConnect defers sending the out-of-sequence packet until the first packet is sent to the client.

The **#out of sequence** column in the **imcadmin -trans** output displays the number of times MaxConnect received an out-of-sequence packet. If the value in the **#out of sequence** column is greater than 10 percent of the number of reads, as the following formula shows, check the user response times:

$$\#OutOfSequence * 10 > (reads * AvgAggr)$$

If response time and MaxConnect CPU utilization are acceptable, do nothing.

If response time and MaxConnect CPU utilization are not acceptable, check the network traffic handled by MaxConnect in the **KB/second** column.

- If the value in **KB/second** is low (less than 100 to 150 kilobytes per second), reduce the number of transports in the **IMCTransports** configuration parameter. If one transport is able to handle the network traffic, you can change **IMCTransports** to 1. Using only one transport eliminates all out-of-sequence packets.
- Determine if the value in the **#out of sequence** column is very high (greater than 20 percent of the number of reads), as the following formula shows:

$$\#OutOfSequence * 20 > (reads * AvgAggr)$$

User response times might improve if you change the application buffer sizes. For more information on the **FetBufSize** and **FetArrSize** global variables and the **FET_BUF_SIZE** environment variable, refer to the [Informix ESQL/C Programmer's Manual](#).

Changing the Number of Transports

You can change the number of transports by one of the following methods:

- Use an editor to change the value of the **IMCTransports** configuration parameter and restart MaxConnect with the **imcadmin -stop** and **-start** options.
- From the main menu of ISA, select Informix, navigate to **Configuration→IMC Config**, click the **Edit** button, change the value of **IMCTransports**, save and restart MaxConnect using the **MaxConnect→Mode** option.

Number of Worker Threads

As [Chapter 4](#) explains, each MaxConnect worker thread manages a queue of packets to send between clients and the database server. The IMCWORKERTHREADS configuration parameter specifies the number of worker threads. The default value of 1 is sufficient for most cases.

Monitoring Worker Threads and Tuning IMCWORKERTHREADS

Monitor the average worker queue length to determine if the current value of the IMCWORKERTHREADS configuration parameter is sufficient for the workload or needs to be larger. Use either of the following tools to monitor the average worker queue length:

- **ISA Status→Worker** option on the MaxConnect server
- **imcadmin→worker**

The output is the same for both tools. The following example shows an excerpt from sample output:

```
WORKER_stats for 900 seconds at 10/07/99 16:54:29 up for 01:04:32,
reset at 10/07/99 16:39:29

...
#dequeues  MaxQlen  AvgQlen      Sends  MaxSpins  AvgSpins
461444      8          1.02      455834  0         0.00
```

Look at the **AvgQlen** column, which shows the average number of packets in each worker queue. If the value of this column is greater than 5, increase the number of worker threads.

Changing the Number of Worker Threads

You can change the number of worker threads by one of the following methods:

- Use an editor to change the value of the IMCWORKERTHREADS configuration parameter and restart MaxConnect with the **imcadmin -start** option.

- Select Informix MaxConnect on the main menu of ISA, navigate to **Configuration→IMC Config**, click the **Edit** button, change the value of IMCWORKERTHREADS, save and restart MaxConnect using the **MaxConnect→Mode** option.

Worker Thread Delay and Packet Aggregation

In OLTP environments, the size of network messages tends to be small. For example, a FETCH request is typically less than 100 bytes. Each MaxConnect worker thread can wait to accumulate multiple packets before it performs a send operation and then sends the aggregated packets to the database server. Packet aggregation significantly reduces the number of operating-system calls to read or write network messages.

The IMCWORKERDELAY configuration parameter specifies the number of microseconds to wait before a send operation. The default delay time is 0. Normally, you do not need to modify the IMCWORKERDELAY configuration parameter because aggregation occurs even when this parameter has a value of 0.

***Important:** If you modify IMCWORKERDELAY, user response times might become longer. Do not modify IMCWORKERDELAY unless you have carefully analyzed the **imcadmin** outputs and user response times.*

The IMCWORKERDELAY configuration parameter might not be available on all hardware platforms. Check the machine notes file to see if IMCWORKERDELAY is available on your platform. For information on the machine notes, see [“Documentation Notes, Release Notes, Machine Notes” on page 10](#) of the Introduction.

Monitoring Packet Aggregation

To monitor statistics for MaxConnect packet aggregation, you can use one of the following methods:

- **imcadmin -aggr, -worker, -trans** options
- **ISA Status→Aggregation, Status→Transports, Status→Worker** options on MaxConnect server

Worker Thread Delay and Packet Aggregation

The output is the same for both the ISA and **imcadmin** options. [Figure 5-1](#) shows sample output.

```

AGGREGATION_stats for 900 seconds at 08/31/99 12:04:46 up for 01:01:43,
reset at 08/31/99 11:49:46
IMC to database aggregation histogram [aggregation count]
  sum      <2      <4      <6      <8      <10     <12     <14     >=14
868432  810128  58089   181     13     10      8       2       1
100.0%  93.3    6.7     0.0     0.0    0.0     0.0     0.0     0.0

database to IMC aggregation histogram [aggregation count]
  sum      <2      <4      <6      <8      <10     <12     <14     >=14
801219  675816  94122  21982  8021   1117   122     24      15
100.0%  84.3    11.7   2.7    1.0    0.1     0.0     0.0     0.0

WORKER_stats for 900 seconds at 08/31/99 12:04:46 up for 01:01:43,
reset at 08/31/99 11:49:46

workers wakeups sleep(s) avg(us) delayed
workers   wakeups   sleep(s)   avg(us)   delayed
  1       814281   827.577776  827       0

#dequeues MaxQlen AvgQlen Sends MaxSpins AvgSpins
886626    14      1.05    868424    0      0.00

arrival interval histogram (micro seconds)
  sum <50 <100 <150 <200 <300 <350 <400 <500 <750 <1000 >=1000
814281 4928 47415 43781 42638 76384 35544 33244 59598 116443 90378 263928
100.0% 0.6 5.8 5.4 5.2 9.4 4.4 4.1 7.3 14.3 11.1 32.4

TRANSPORT_stats for 900 seconds at 08/31/99 12:04:46 up for 01:01:43,
reset at 08/31/99 11:49:46

Write statistics for 7 transports:
  writes      total  avg  max  avg  max  partial  KB writes
              KB bytes bytes aggr aggr  writes  /sec  /sec
868434      328254  387 4700 1.1 14    0  364.73  964

Read statistics for 7 transports:
  reads      total  avg  max  avg  max  partial  KB reads
              KB bytes bytes aggr aggr  reads  /sec  /sec
801219      795717 1016 22698 1.3 15    0  884.13  890

#transport #out of
  spins    sequence
  0        72253
    
```

Figure 5-1
Aggregation,
Worker, and
Transport Statistics
in ISA and
imcadmin Output

To monitor the effectiveness of packet aggregation

1. Even when IMCWORKERDELAY is 0, some aggregation occurs, as the aggregation histograms in [Figure 5-1](#) show. Add the percentages under the columns that indicate multiple packets are contained in the message sent.

For example, [Figure 5-1](#) shows nonzero values for the <4, <6, <8, and <10 columns in the database to IMC aggregation histogram portion of the **imcadmin -aggr** output. Packet aggregation occurs for about 15 percent of the messages that the database server sent to MaxConnect.

$$\begin{aligned} \text{Percent aggregated} &= 11.7 + 2.7 + 1.0 + 0.1 \\ &= 15.8\% \end{aligned}$$

2. Look at the following columns in the **imcadmin -worker** output:

- **AvgQlen** column

For the example in [Figure 5-1](#), the **AvgQlen** column shows a value of 1.05, which is a low average queue length and indicates that packet aggregation is not frequent.

Large values in the **AvgQlen** column indicate more effective packet aggregation. If response times are good, you might try to improve packet aggregation. For more information, refer to [“Tuning IMCWORKERDELAY” on page 5-21](#).

- The **delayed** column indicates the number of packets that MaxConnect can aggregate if IMCWORKERDELAY is greater than 0.

In [Figure 5-1](#), the **delayed** column shows a value of 0. This value increases if IMCWORKERDELAY is greater than 0.

3. Look at the **avg aggr** columns in the `Write statistics` and `Read statistics` portions of the `imcadmin -trans` output.

Figure 5-2 shows an example where packet aggregation is more effective for messages that the database server sends to MaxConnect than from MaxConnect to the database server.

- The **avg aggr** column in the `Write statistics` portion shows the average number of client network messages that MaxConnect combined into one larger network message to be sent to the database server. The **avg aggr** column in **Figure 5-2** shows a value of 1.0, which is a low value and indicates that packet aggregation from MaxConnect to the database server is not frequent.

If response times are good, you might try to improve the packet-aggregation rate to reduce CPU utilization. For more information, refer to “[Tuning IMCWORKERDELAY](#)” on [page 5-21](#).

- The **avg aggr** column in the `Read statistics` portion shows the average number of database server network messages that have been combined into one larger network message to be sent to MaxConnect. In **Figure 5-2**, this column shows a value of 6.1, which is a good aggregation rate and indicates that the CPU processing time has been reduced in the database server.

```
TRANSPORT_stats for 72257 seconds at 02/08/00 13:18:08 up for 21:38:44,
reset at 02/07/00 17:13:51
```

```
Write statistics for 4 transports:
```

writes	total	avg	max	avg	max	partial	KB	writes
		KB	bytes	aggr	aggr	writes	/sec	/sec
5213	1565	307	790	1.0	2	0	0.76	2

```
Read statistics for 4 transports:
```

reads	total	avg	max	avg	max	partial	KB	reads
		KB	bytes	aggr	aggr	reads	/sec	/sec
859	11985	14288	32376	6.1	15	1123	5.84	0

```
#transport #out of
spins sequence
0 0
```

Figure 5-2
Aggregation
Statistics in
`imcadmin -trans`
Output



Tuning IMCWORKERDELAY

Important: If you modify the IMCWORKERDELAY configuration parameter, user response times might become longer. Do not modify IMCWORKERDELAY unless you have carefully analyzed the *imcadmin* outputs and user response times.

To determine if you need to modify IMCWORKERDELAY

1. Monitor user response times.

If user response times are low, you can tune IMCWORKERDELAY to try to improve the CPU utilization of the database server.

If the CPU utilization is high for the database server and MaxConnect, perform steps 2 and 3.

2. Look at the **avg aggr** columns in the `Write statistics` and `Read statistics` portions of the output of **imcadmin -trans** or **ISA Status→Transports** on the MaxConnect server.

[Figure 5-3](#) shows an example where little packet aggregation occurs for messages that MaxConnect sends to the database servers.

The **avg aggr** column in the `Write statistics` portion shows the average number of client network messages that MaxConnect combined into one larger network message to be sent to the database server. [Figure 5-3](#) shows a value of 1.0 for this column. Step 3 gives guidelines on how to improve this aggregation.

- Look at the histograms of arrival rates in **imcadmin -worker**.
Sum the values in the **<50, <100, <150, and <200** columns in the arrival interval histogram portion of the **imcadmin -worker** output in **Figure 5-3**. This calculation shows that about 15 percent of messages sent by the database server to MaxConnect occur in less than 200 microseconds.

$$\begin{aligned} \text{Arrivals} < 200 \text{ microseconds} &= 0.8 + 5.4 + 5.1 + 4.9 \\ &= 15.3\% \end{aligned}$$

From this calculation, you can conclude that MaxConnect can aggregate about 15 percent additional packets if you set **IMCWORKERDELAY** to 200 microseconds. However, this setting might have an adverse effect on average response times because individual client messages can be delayed by up to 200 microseconds.

Figure 5-3
MaxConnect
Transport and
Worker Statistics

```

TRANSPORT_stats for 2051 seconds at 01/26/00 21:22:32 up for 00:34:12,
reset at 01/26/00 20:48:21

Write statistics for 4 transports:
  writes      total  avg  max  avg  max  partial  KB  writes
              KB bytes bytes aggr aggr  writes   /sec /sec
  5213        1565  307  790  1.0  2     0       0.76  2

Read statistics for 7 transports:
  reads      total  avg  max  avg  max  partial  KB  reads
              KB bytes bytes aggr aggr  reads   /sec /sec
  859        11985 14288 32376 6.1  15   1123   5.84  0

#transport  #out of
  spins     sequence
  0         0

WORKER_stats for 900 seconds at 12/06/99 15:02:59 up for 01:36:54,
reset at 12/06/99 14:47:59

workers wakeups sleep(s) avg(us) delayed
workers  wakeups  sleep(s)  avg(us)  delayed
  1      814010  831.522504  831      0

#dequeues  MaxQlen  AvgQlen      Sends  MaxSpins  AvgSpins
  878578    26      1.04      863248  0         0.00

arrival interval histogram (micro seconds)
  sum  <50  <100  <150  <200  <300  <350  <400  <500  <750  <1000  >=1000
814011  6187  44170  41540  40034  73059  33755  31794  57691  11804  87020  280707
100.0%  0.8   5.4   5.1   4.9   9.0   4.1   3.9   7.1   4.5   10.7   34.5
    
```


Load Balancing Across Multiple Instances

Informix recommends that you do not install multiple instances of MaxConnect unless you need them. If you have a natural division of work among the instances you have installed, use it to distribute the load. For example, you might distribute client sessions geographically, with clients from one location sending requests through one MaxConnect instance and clients at another location sending through another instance.

If there is no natural work-load division, use the group feature with the connection redirection option to distribute the load in a balanced way across the MaxConnect instances. Using this feature, you can have the client sessions go to each MaxConnect instance in round-robin order to avoid potential CPU resource bottlenecks.

UNIX

Establishing Server Groups on UNIX

To establish a group, you must create it in the **sqlhosts** file of the client by defining a group and adding each MaxConnect instance to the group.

For example, configure three listener threads on the same database server, one for each MaxConnect instance that you want to include in the group. In the following example these listeners are designated as **max100t**, **max200t**, and **max300t**.

dbservername	nettype	hostname	servicename	options
maxserv	group	-	-	c=1
max100t	ontlitcp	apples	57004	g=maxserv
max200t	ontlitcp	apples	65004	g=maxserv
max300t	ontlitcp	apples	55004	g=maxserv

Set the **INFORMIXSERVER** client-side environment variable to the value of the group name. In this example the group name is `maxserv`.

Setting	Result
<code>c=1</code>	The client-side communications library randomly picks one of the three MaxConnect instances to connect to. Choose this option if you want a reasonably uniform distribution of clients over MaxConnect instances.
<code>c=0</code>	The client-side communications library will use a round-robin algorithm within the group <code>maxserv</code> and try to connect first to the first MaxConnect instance listed in the group (max100t), next to the MaxConnect instance listening on max200t , and last to the MaxConnect instance listening on max300t .

For more information on the group option, see the client/server chapter of the [Administrator's Guide for Informix Dynamic Server 2000](#).

WIN NT

Establishing Server Groups on Windows NT

The recommended tool for preparing SQLHOSTS is the Client Configuration tool, which is provided as part of "Informix DB Administrator."

After you install the Client Configuration tool, follow these steps to prepare SQLHOSTS information:

1. Start the Client Configuration tool.
2. Double-click the **Network** icon.
3. Select the **Servers** tab in the Network Properties dialog box.
4. Add your database servers.
5. Select the **Server Groups** tab in the Network Properties dialog box.
6. Create your database server groups.
7. Add your database servers to the groups.

You must also set user login information for each user and host.

1. Start the Client Configuration tool.
2. Double-click the **Login** icon.
3. Provide user information in the Login Properties dialog box.

If you currently use **regedt32** for this task, you can continue to use that utility. For more information on using **regedt32** to establish server groups, see [Appendix A](#).

Providing for MaxConnect Failure

To redirect a client application to another MaxConnect instance if a failure occurs, use the connection-redirection option of the group feature. The example in “[Load Balancing Across Multiple Instances](#)” on page 5-23 applies in this case also.

In a situation where the client’s communications library cannot connect to one member of the group, for example, if the MaxConnect instance rejects the connect request because of a shortage of resources or the MaxConnect instance is not up and running, the library will try another member of the group **maxserv**.

You can tune the failover-detection-timeout period and number of attempts to connect by setting the environment variables **INFORMIXCONRETRY** and **INFORMIXCONTIME** on the client computer. For further information on these environment variables, see the [Informix Guide to SQL: Reference](#).

Establishing Server Groups Using regedt32

WIN NT

SQLHOSTS Registry Key

When you install Informix products, the **setup** program creates the following key in the Windows NT registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS
```

This branch of the HKEY_LOCAL_MACHINE subtree stores the **sqlhosts** information. Each key on the SQLHOSTS branch is the name of a database server. When you click the database server name, the registry displays the values of the HOST, OPTIONS, PROTOCOL, and SERVICE fields for that particular database server.

Each computer that hosts a client must include the connectivity information either in the SQLHOSTS registry key or in a central registry.

Modifying the SQLHOSTS Connectivity Information

To add or modify the SQLHOSTS connectivity information, you can use the Windows NT program, **regedt32**.



Important: Use extreme caution with **regedt32**. If you make mistakes when editing the registry, you can destroy the configuration, not only of your Informix products, but of your other applications.

To modify dobserver connectivity information for members of the group

1. Run **regedt32**.
2. In the Registry Editor window, select the window for the HKEY_LOCAL_MACHINE subtree.
3. Click the folder icons to select the \SOFTWARE\INFORMIX\SQLHOSTS branch.

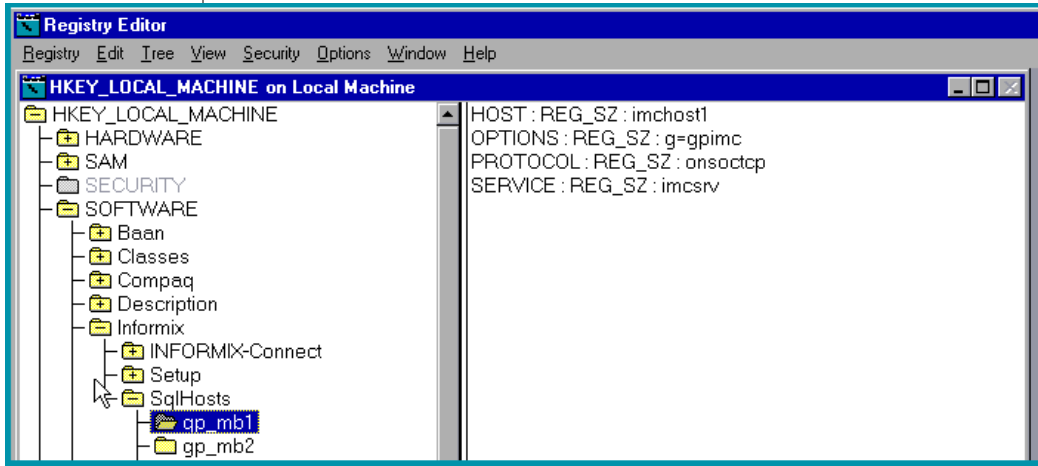
(These instructions assume that your client is already configured to connect to the database server. If your client is not already configured, select the SQLHOSTS key, choose **Edit→Add Key**, and enter the name of the MaxConnect server in the Key Name dialog box. Leave the Class dialog box blank and click **OK**. Then add values for HOST, OPTION, PROTOCOL, and SERVICE.)
4. Select the key for the first group member.
5. In the right-hand pane, select one of the fields and choose **Edit→String**.
6. In the String Editor dialog box, enter the appropriate value for the field as the following table shows. This example is also illustrated in [Figure A-1 on page A-3](#).

HOST:imchost1	Change the host from the host of the database server to the host of the MaxConnect instance that the client will use to connect to the database server.
OPTION:g=gpimc	Add the g option and choose a group name for the group of MaxConnect instances that this client uses.
PROTOCOL:onsoctcp	You do not need to change this value unless you installed the MaxConnect instance on a computer that requires a protocol different from that used by the database server.
SERVICE:imcsrv	Change the service name to the service name used by the MaxConnect instance to listen for client connections.

7. Repeat steps 5 and 6 for each of the fields that you must modify.

Figure A-1 illustrates the location and contents of the SQLHOSTS registry key.

Figure A-1
sqlhosts Information (server connections) in the Windows NT Registry



After you modify the connectivity information, you must make a server group that includes the MaxConnect instances and add a subkey for each member of the group.

To add MaxConnect server group information

1. With the SQLHOSTS key selected, choose **Edit→Add Key**.
2. In the Add Key dialog box, enter the name of the MaxConnect instance in the Key Name dialog box.

In the example illustrated in [Figure A-2 on page A-5](#), the new group name is **gpimc**.

Leave the Class dialog box blank. Click **OK**.

3. Select the key that you just made (**gpimc**).
4. Choose **Edit→Add Value**.
5. In the Add Value dialog box, enter one of the fields of the **sqlhosts** information (HOST, OPTIONS, PROTOCOL, SERVICE) in the Value Name dialog box.

Do not change the Data Type dialog box. Click **OK**.

6. In the String Editor dialog box, type the value for the field that you selected. Enter the following values for each field:

```
HOST      -
OPTIONS   i=unique-integer-value
PROTOCOL  group
SERVICE  -
```

Each server group must have an associated identifier value (i=) that is unique among all servers in your environment. In the case of MaxConnect servers, this is the MaxConnect identifier number. Enter a minus (-) for the HOST and SERVICE fields to indicate that you are not assigning specific values to those fields.

Click **OK**.

7. Repeat steps 5 and 6 for each of the four fields.
8. Add a subkey for each member of the group.

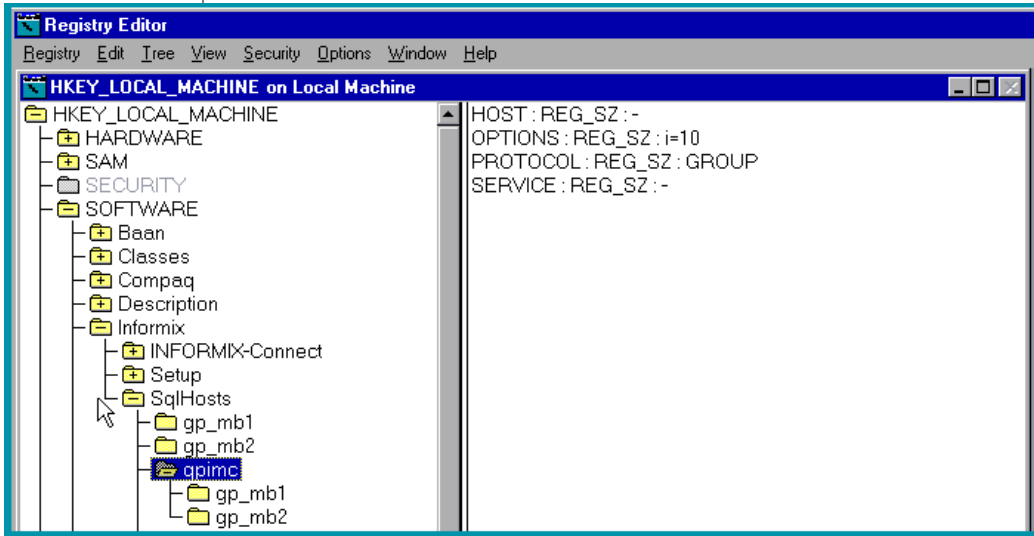
1. Select the group name.
2. Choose **Edit→Add Key**
3. Type in the name of a member of the group.
4. Leave the Class field empty and click **OK**.
5. Repeat this step for each member of the group.

In the example illustrated in [Figure A-2 on page A-5](#), subkeys have been added for group members **gp_mb1** and **gp_mb2**.

9. Exit from the Registry Editor.
10. On the database server, add a DBSERVERALIASES entry for each member of the group. In this example, add entries to the DBSERVERALIASES list for **gp_mb1** and **gp_mb2**.

Modifying the SQLHOSTS Connectivity Information

Figure A-2
sqlhosts Information (server groups) in the Windows NT Registry



Index

A

Admin thread
 configuring 3-11
 port number 3-11
 Administration utility. *See*
 imcadmin utility.
 Aggregation
 delay 2-17, 5-17, 5-19, 5-22
 overview 1-6

B

Boldface type Intro-5

C

Comment icons Intro-6
 configimc
 command 3-6
 script 3-6
 Configuration
 alternatives 2-3
 file 3-7
 file example 2-18
 Configuration parameter
 IMCWORKERDELAY 5-17, 5-19,
 5-21, 5-22
 IMCWORKERTHREADS 5-16
 Configuration parameters
 affecting network free buffer 5-9
 IMCLOG 2-17
 IMCTRAMSPORTS 2-17
 IMCTRAMSPORTS 5-12,
 5-14, 5-15
 IMCWORKERDELAY 2-17

IMCWORKERTHREADS 2-17
 NETTYPE 2-12, 4-9, 5-8, 5-9, 5-10,
 5-11
 specifying 2-17
 Configuring
 as standalone 3-9
 client computers 2-18
 database server computer 2-9
 MaxConnect host computer 2-14
 using script 3-6
 Connection multiplexing
 illustrated 1-5, 2-4, 2-5
 in MaxConnect 1-6
 sqlmux protocol 2-13
 Connection protocol 2-13
 Connections, server-to-server 2-6
 Contact information Intro-11
 CPU utilization
 effect of multiple protocols 5-4
 improved response time 5-6
 reducing on database server 5-4
 reducing with
 IMCWORKERDELAY 5-21
 reducing with packet
 aggregation 5-6, 5-20
 CPU virtual processor type 2-13

D

Database server groups A-3
 DBSERVERALIASES 2-11
 Dependencies, software Intro-4
 Distributed queries 2-6
 Documentation notes Intro-10
 Documentation, types of
 documentation notes Intro-10
 machine notes Intro-10

online manuals Intro-9
 printed manuals Intro-9
 related reading Intro-11
 release notes Intro-10

E

Environment variables
 correct settings 3-16
 IMCADMIN 2-15, 2-20
 IMCONFIG 2-15
 IMCSERVER 2-15
 INFORMIXDIR 2-15
 INFORMIXSERVER 2-11, 2-15
 INFORMIXSQLHOSTS 2-15
 on MaxConnect host 2-14
 Error messages
 location of log file 2-17
 monitoring 4-31

F

Failover, MaxConnect
 instances 5-25
 Feature icons Intro-7
 Formula, threshold for free network
 buffers 5-9

G

Group feature 5-23

I

Icons
 feature Intro-7
 important Intro-6
 platform Intro-7
 product Intro-7
 Tip Intro-6
 Warning Intro-6
 IMCADMIN environment
 variable 2-15, 2-20
 imcadm utility
 configuring 2-19
 interactive mode 4-10
 monitoring transports 5-11

options
 -aggr 4-21
 -log 4-25
 -poll 4-24
 -reset 5-10
 -ses 5-10
 -status 4-23
 -trans 5-11, 5-13, 5-14, 5-15
 overview 4-9
 syntax 4-11
 IMCONFIG environment
 variable 2-15
 IMCLOG configuration
 parameter 2-17
 IMCSERVER environment
 variable 2-15
 IMCTransports configuration
 parameter 4-6
 changing with ISA 5-15
 description 2-17
 initial setting 3-8
 monitoring 5-12, 5-14
 monitoring with imcadm 5-13
 tuning 5-15
 tuning for Wide-Area
 Network 5-8
 IMCWORKERDELAY
 configuration parameter 2-17,
 5-17
 IMCWORKERDELAY
 configuration parameter
 effect on aggregation 5-19
 effect on response time 5-17,
 5-21, 5-22
 monitoring 5-17
 reducing CPU utilization 5-21
 tuning 5-22
 IMCWORKERTHREADS
 configuration parameter 2-17
 monitoring 5-16
 imc.log file 2-17
 imc.tar file 3-3
 Important paragraphs, icon
 for Intro-6
 Informix Server Administrator
 (ISA)
 administering MaxConnect 4-8
 changing IMCTransports
 configuration parameter 5-15

monitoring MaxConnect 4-8
 monitoring MaxConnect
 transports 5-11
 monitoring network buffers 5-10
 monitoring sessions 5-10
 INFORMIXDIR environment
 variable 2-15
 INFORMIXSERVER environment
 variable 2-15
 INFORMIXSQLHOSTS
 environment variable
 changing network protocols 5-5
 description 2-15
 Installation
 directory 3-3
 script 3-4
 script requirements 3-5
 installimc command 3-4
 ipcstr protocol 2-9
 ISA. *See* Informix Server
 Administrator.

L

Large data objects 1-4
 Listener thread
 configuring with script 3-10
 description of 4-5
 supported protocols 3-10
 Load balancing 5-23
 Log file
 creating 3-8
 location 2-17
 naming 3-8
 statistics 4-25

M

Machine notes Intro-10
 MaxConnect
 administering 4-3
 architecture 1-5
 configuring 2-3
 described 1-3
 features 1-4
 improving performance 5-3
 installing 3-3
 monitoring 4-3

starting 4-7
 stopping 4-7
 uninstalling 3-4
 Maximum Transfer Unit (MTU) 5-12, 5-14
 Memory, network buffer pool 5-9
 Monitoring
 network buffers 5-10
 transports 5-11, 5-13, 5-14, 5-15
 with imcadmin -ses 5-10
 with imcadmin -trans 5-11, 5-13, 5-14, 5-15
 with onstat utility 4-25
 Multiple MaxConnect instances 2-16
 Multiplexed connections
 MaxConnect 1-6
 sqlmux and MaxConnect 5-6

N

NET virtual processor type 2-13
 NETTYPE configuration
 parameter 2-12
 buffers for sessions 5-11
 network free buffer 5-9
 sessions 4-9, 5-8, 5-10
 Network
 changing protocols 5-5
 common buffer pool 5-9
 number of poll threads 5-4
 sqlmux connections 5-6
 support of multiple protocols 5-4
 Network buffers
 NETTYPE configuration
 parameter 5-11

O

ONCONFIG file 2-11
 Online manuals Intro-9
 onstat -g imc
 monitoring network buffers 5-10
 monitoring transports 5-8, 5-12, 5-13
 onstat utility, monitoring
 MaxConnect with 4-25
 Operating system parameters 2-18

P

Packet aggregation
 good rate 5-20
 monitoring 5-18, 5-19, 5-20, 5-21
 reducing CPU utilization 5-6, 5-20, 5-21
 statistics 4-21
 tuning 5-21, 5-22
 Parameters, operating-system 2-18
 Performance, improving 5-3
 Platform icons Intro-7
 Poll threads
 description of 4-6
 network protocol 5-4
 on database server 4-6
 recommended number of 2-13
 reducing number of 5-4
 statistics 4-24
 Printed manuals Intro-9
 Process priority of MaxConnect 5-6
 Product icons Intro-7
 Protocol
 ipcstr 2-9
 Protocol, connection 2-13

R

Receive threads 4-6
 regedt32 program and sqlhosts registry A-2
 Related reading Intro-11
 Release notes Intro-10
 Response time
 effect of IMCWORKDELAY 5-17
 effect of IMCWORKERDELAY 5-22
 improving with packet aggregation 5-6
 RUN_AS_ROOT script 3-17

S

Serial number 3-5
 Session ids 4-5
 Sessions
 NETTYPE configuration
 parameter 4-9, 5-8, 5-10

setup program, and sqlhosts registry A-1
 Software dependencies Intro-4
 sqlhosts file
 configuration script 3-9
 database server 2-9
 on client computer 2-18
 on MaxConnect computer 2-15
 reducing CPU utilization 5-5
 view or change 4-8
 sqlhosts on Windows NT 2-19, 3-15
 sqlmux
 reduced network connections 5-6
 using with MaxConnect 5-6
 sqlmux protocol 2-13
 Standalone configuration 3-9
 Starting and Stopping
 MaxConnect 4-7
 Starting MaxConnect 4-7
 Statistics
 log file 4-25
 packet aggregation 4-21
 poll threads 4-24
 status of MaxConnect instance 4-23
 Stopping MaxConnect 4-7
 Syntax
 configimc command 3-6
 conventions, description of Intro-7
 installimc command 3-4
 System requirements
 database Intro-4
 software Intro-4

T

Tip icons Intro-6
 Transports
 defined 4-6
 IMCTRANSPORTS configuration
 parameter 5-12, 5-14, 5-15
 specifying number of 2-17

U

Uninstalling MaxConnect 3-4

UNIX

- establishing server groups
on 5-23

Users

- setting number of 2-13
- total number of 2-13

Utilities

- dbaccess 2-10
- imcadmin 4-9
- ISA 4-8
- oncheck 2-10
- onstat 4-25

V

Virtual processor class 2-13

W

Warning icons Intro-6

Wide-Area Network (WAN)

- and MaxConnect 5-7

- network routers 5-7

- tuning IMCTransports

 - configuration parameter 5-8

Windows NT

- establishing server groups
on 5-24, A-1

- modifying sqlhosts 2-19, 3-15

Worker queue 4-5

Worker threads

- description of 4-5

- monitoring 5-16

- parameters 2-17, 5-16

- specifying number of 2-17