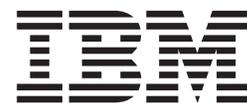


# IMS TCP/IP OTMA Connection Programmer's Reference





# IMS TCP/IP OTMA Connection Programmer's Reference

**Note**

Before using this information and the product it supports, be sure to read the information in "Notices" on page v.

**ITOC Programmer's Reference**

© Copyright International Business Machines Corporation 1997. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Notices</b> . . . . .	v
Reader comment form . . . . .	vi
<b>Trademarks</b> . . . . .	vii
<b>Chapter 1. Conversational support</b> . . . . .	1
OTMA conversational protocols . . . . .	1
ITOC conversational protocols . . . . .	2
<b>Chapter 2. Security support</b> . . . . .	9
<b>Chapter 3. IMS TCP/IP OTMA Connection commands</b> . . . . .	11
CLOSEHWS . . . . .	11
OPENDS. . . . .	11
OPENPORT . . . . .	12
SETRACF . . . . .	12
STOPCLNT. . . . .	12
STOPDS. . . . .	13
STOPPORT . . . . .	13
VIEWDS . . . . .	14
VIEWHWS . . . . .	15
VIEWPORT. . . . .	15
<b>Chapter 4. IMS client for Java</b> . . . . .	17
Contents of JAVASAMP.exe . . . . .	17
Running the sample Java client . . . . .	18
Running the client as a Java application . . . . .	18
Running a client applet locally . . . . .	18
Running the client as a remote applet . . . . .	18
<b>Chapter 5. User message exits for ITOC</b> . . . . .	19
EZAIMSO0 user message exit . . . . .	19
HWSWEB00 user message exit . . . . .	20
HWSSMPL0 user message exit . . . . .	20
<b>Readers' Comments — We'd Like to Hear from You.</b> . . . . .	21



---

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, NY 10594  
U.S.A.

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is as your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
W92/H3  
555 Bailey Avenue  
P.O. Box 49023  
San Jose, CA 95161-9023  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Reader comment form

A reader comment form is included with the hardcopy version of this book. If you would like to fill out the online version of the reader comment form, please link here or go to <http://www.software.ibm.com/data/rcf/>.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

IBM	IMS
MVS	OS/2
RACF	

Windows and Windows NT are registered trademarks of Microsoft Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.



# Chapter 1. Conversational support

A conversational program is a message processing program (MPP) that processes transactions made up of several steps. The MPP does not process the entire transaction at once.

The conversational support for IMS TCP/IP OTMA Connection includes having conversational transactions that let you retain uninterrupted connection (continuity) for messages coming from a given client. Typically, a conversation is terminated when the message is sent and dequeued and the application program has placed blanks in the SPA, or the conversation is terminated when a COMMIT CONFIRMED message is received from the client. For conversational support for ITOC, conversations require a send-then-commit mode and are nonrecoverable.

This section describes and shows various conversational protocols.

## OTMA conversational protocols

- Send-then-commit, sync level=none

The send-then-commit flow (see Figure 1) sends IMS output before IMS completes synchronization-point (hereafter referred to as sync-point) processing. To use the send-then-commit flow, specify Commit Mode 1 in the state-data section of the message prefix. This sample flow assumes the following:

- The transaction pipe is not synchronized.
- The synchronization level is specified as NONE in the state-data section of the message prefix. Therefore, IMS does not request a response (an ACK) when sending output.

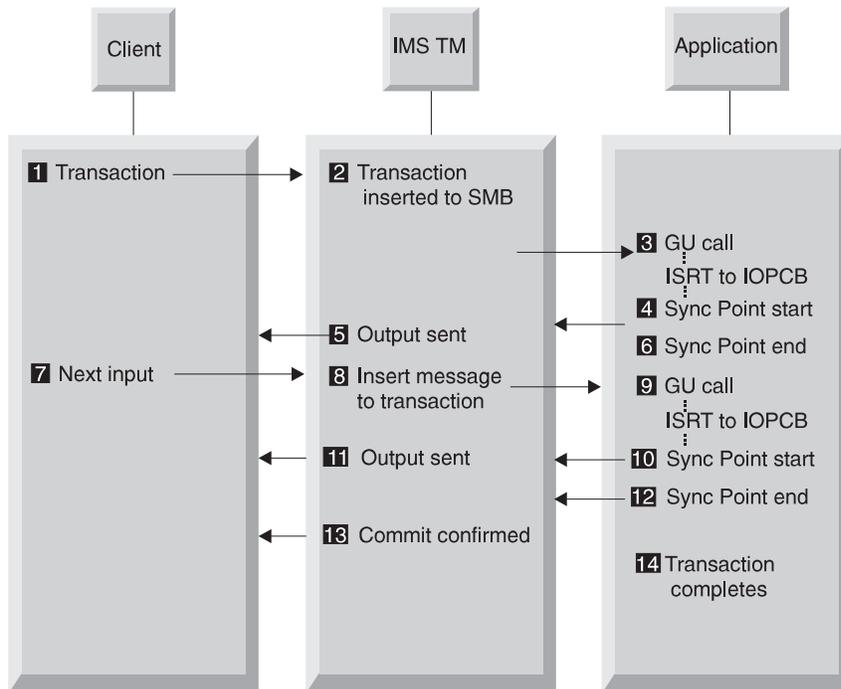


Figure 1. Send-then-commit flow

- Send-then-commit, sync level=confirm

The send-then-commit flow (see Figure 2) assumed no synchronization for the transactions as they are processed by IMS. This section shows a flow in which all transactions are confirmed as they are received (each message requests a response). The sample flow shown in the following figure assumes the following:

- Commit Mode 1 is specified in the state-data section of the message prefix.
- The transaction pipe is not synchronized.
- The synchronization level is specified as Confirm in the state-data section.

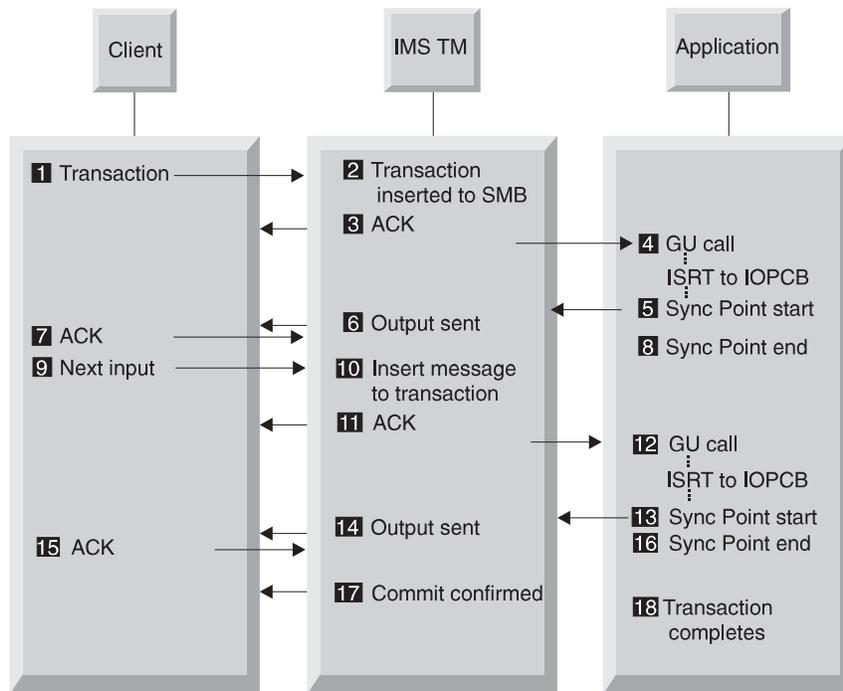


Figure 2. Send-then-commit flow

## ITOC conversational protocols

- Send-then-commit, sync level=none, transaction terminated from the program

The send-then-commit flow (see Figure 3 on page 3) sends IMS output before IMS completes sync-point processing. To use the send-then-commit flow, specify Commit Mode 1 in the state-data section of the message prefix. This sample flow assumes the following:

- The transaction pipe is not synchronized.
- The synchronization level is specified as NONE in the state-data section of the message prefix. Therefore, IMS does not request a response (an ACK) when sending output.
- The transaction is terminated from the program.
- ITOC will close the socket as soon as Commit confirmed has been sent by IMS.

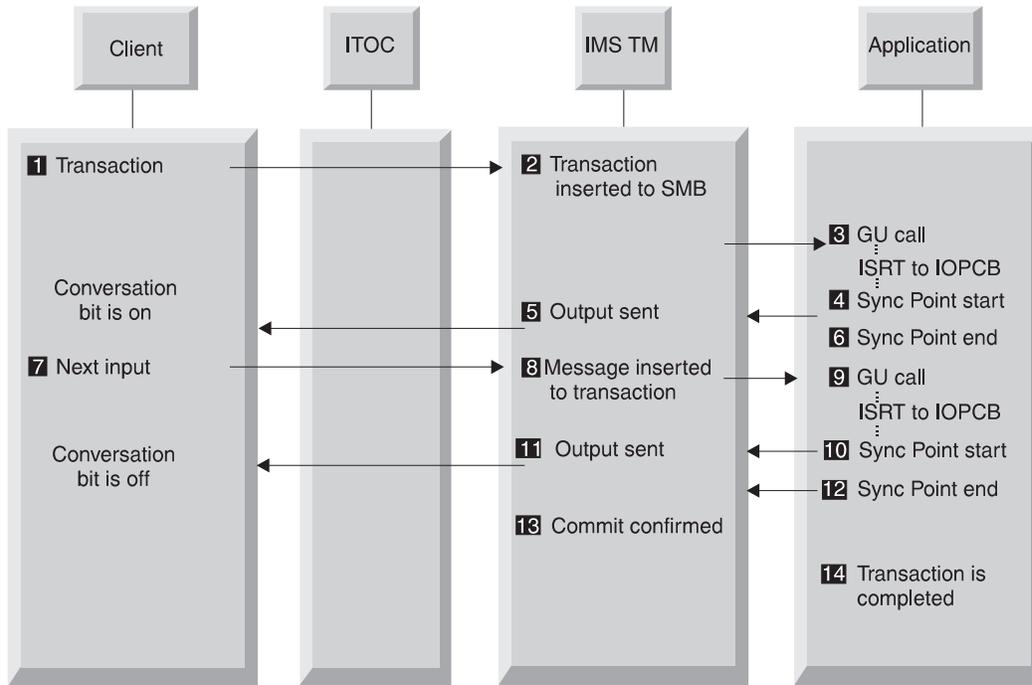


Figure 3. Send-then-commit flow

- Send-then-commit, sync level=none, transaction is terminated from client  
 The send-then-commit flow (see Figure 4 on page 4) sends IMS output before IMS completes sync-point processing. To use the send-then-commit flow, specify Commit Mode 1 in the state-data section of the message prefix. This sample flow assumes the following:
  - The transaction pipe is not synchronized.
  - The synchronization level is specified as NONE in the state-data section of the message prefix. Therefore, IMS does not request a response (an ACK) when sending output.
  - The transaction is terminated from client.

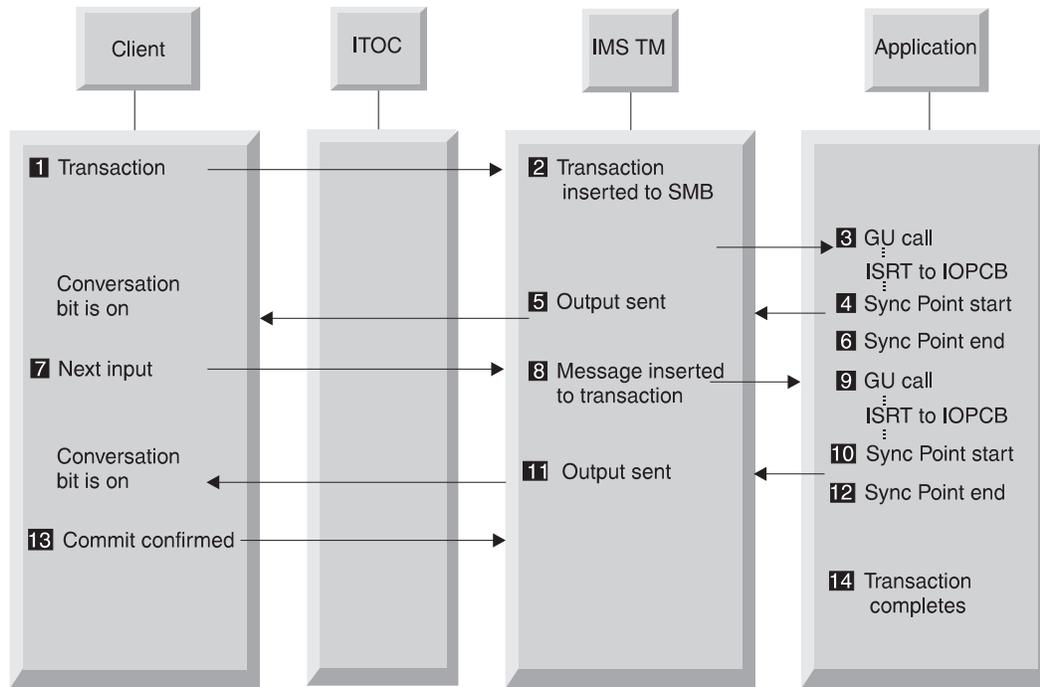


Figure 4. Send-then-commit flow

- Send-then-commit, sync level=confirm, ACK response

The send-then-commit flow (see Figure 5 on page 5) assumed no synchronization for the transactions as they are processed by IMS. This section shows a flow in which all transactions are confirmed as they are received (each message requests a response). The sample flow shown in the following figure assumes the following:

- Commit Mode 1 is specified in the state-data section of the message prefix.
- The transaction pipe is not synchronized.
- The synchronization level is specified as Confirm in the state-data section.
- ACK can be replied to by a remote workstation before the check response requested bit.

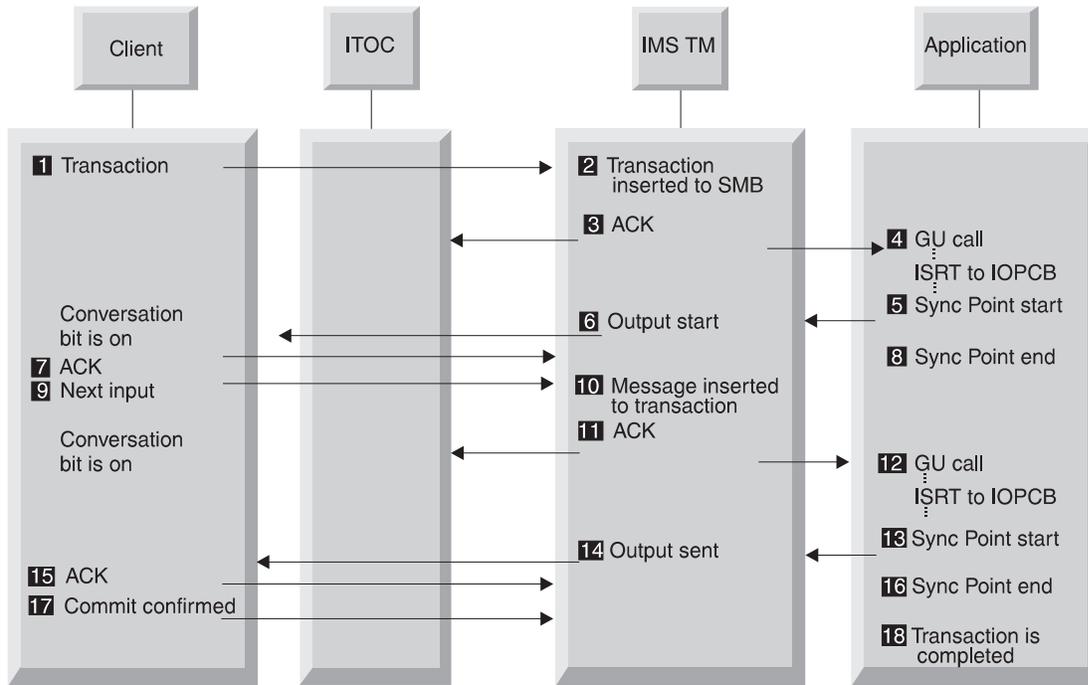


Figure 5. Send-then-commit flow

- Send-then-commit, sync level=confirm, NAK response

The send-then-commit flow (see Figure 6 on page 6) assumed no synchronization for the transactions as they are processed by IMS. This section shows a flow in which all transactions are confirmed as they are received (each message requests a response). The sample flow shown in the following figure assumes the following:

- Commit Mode 1 is specified in the state-data section of the message prefix.
- The transaction pipe is not synchronized.
- The synchronization level is specified as Confirm in the state-data section.
- NAK can be replied to by either ITOC or a remote workstation before the check requested bit.
- If the client forgets to send the NAK/ACK before it closes the socket, ITOC will send the NAK to IMS and it will cause a U0119 abend.

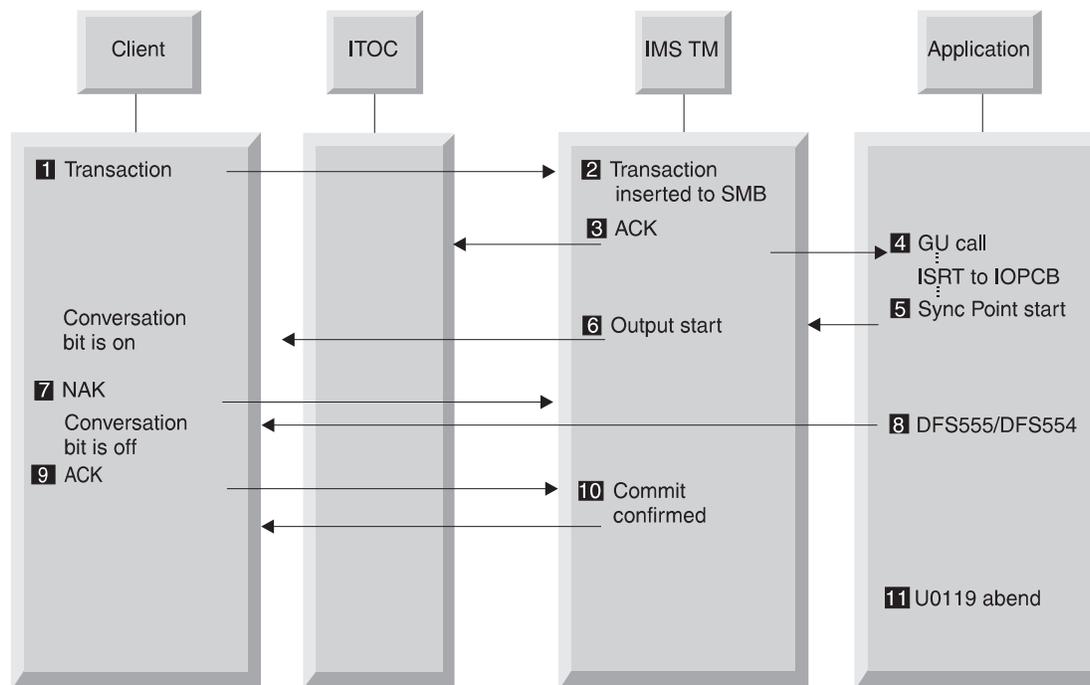


Figure 6. Send-then-commit flow

- Send-then-commit, sync level=confirm, nonpersistent socket, ACK response  
 The send-then-commit flow (see Figure 7 on page 7) assumed no synchronization for the transactions as they are processed by IMS. This section shows a flow in which all transactions are confirmed as they are received (each message requests a response). The sample flow shown in the following figure assumes the following:
  - Commit Mode 1 is specified in the state-data section of the message prefix.
  - The transaction pipe is not synchronized.
  - The synchronization level is specified as Confirm in the state-data section.
  - The server token in the state-data should be used to send the next input message.
  - ACK can be replied to by a remote workstation before the check response requested bit.
  - ITOC will close the socket after it receives the ACK.
  - If the client goes away and the conversation bit is still on before a Commit Confirm is sent, the STOPCLNT command can be used to clear the control block.

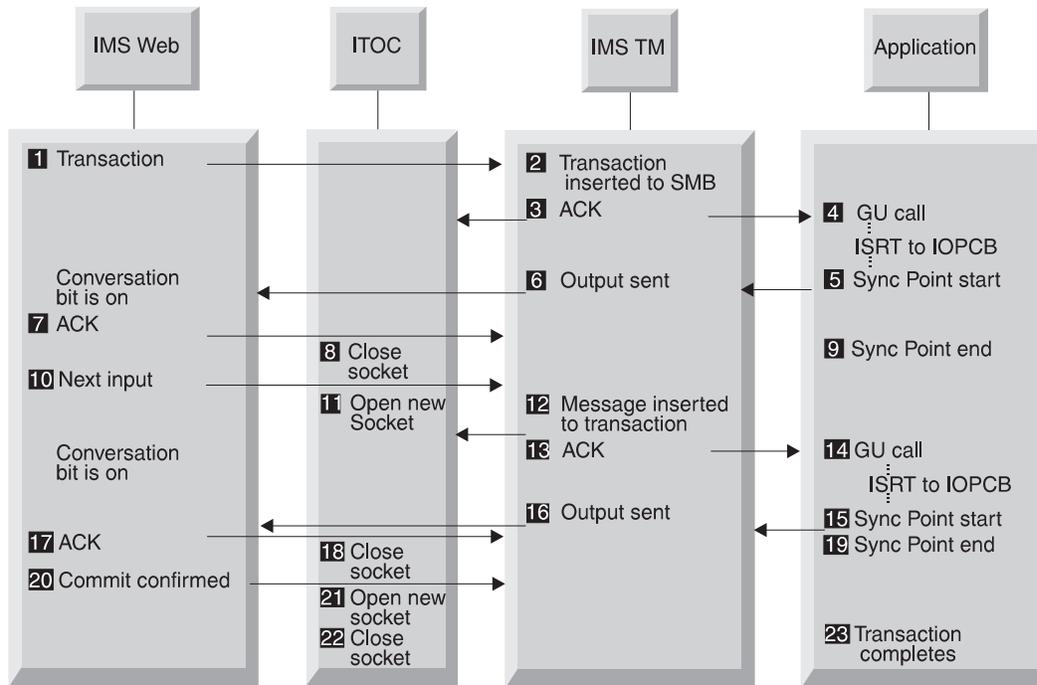


Figure 7. Send-then-commit flow

- Send-then-commit, sync level=confirm, nonpersistent socket, NAK response  
 The send-then-commit flow (see Figure 8 on page 8) assumed no synchronization for the transactions as they are processed by IMS. This section shows a flow in which all transactions are confirmed as they are received (each message requests a response). The sample flow shown in the following figure assumes the following:
  - Commit Mode 1 is specified in the state-data section of the message prefix.
  - The transaction pipe is not synchronized.
  - The synchronization level is specified as Confirm in the state-data section.
  - The server token in the state-data should be used to send the next input message.
  - NAK can be replied to by a remote workstation before the check response requested bit.
  - ITOC will close the socket after it receives the NAK. The client control block will still be present in ITOC.
  - IMS sends the DFS555/554 message.
  - ITOC will ACK the DFS555/554 message and discard it.
  - IMS will issue a Commit Confirm to ITOC and ITOC will discard it.
  - If the client forgets to send the NAK/ACK before it closes the socket, ITOC will send the NAK to IMS and it will cause a U0119 abend.

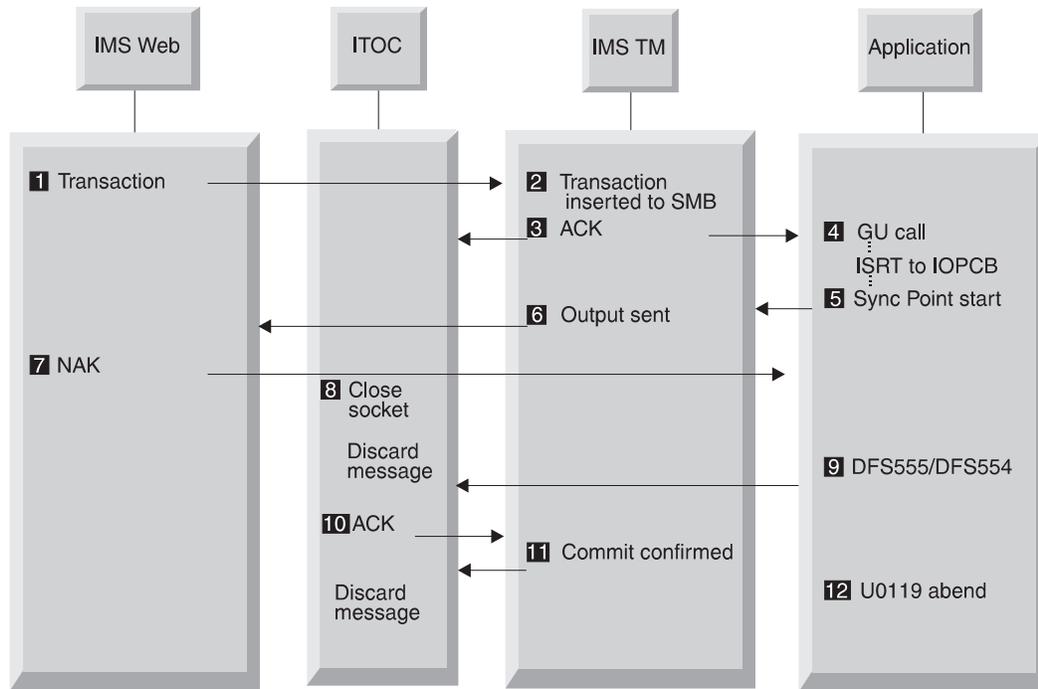


Figure 8. Send-then-commit flow

**Requirement:** To support conversations, the user exit routine must also support the above protocol. Check that the EXPREA\_CONVERS bit is on.

---

## Chapter 2. Security support

ITOC allows security support by checking the RACF flag. There are two ways to activate the RACF flag:

- You can set the flag in the configuration file, HWSCFG00, by setting the flag to Y as follows:

```
HWS ID=HWS01 RACF=Y
```

- You can also use the HWS command, SETRACF, to set the flag as follows:

```
SETRACF ON
```

To check the setting of the RACF flag, you can issue the VIEW HWS command. After you issue this command, you should see: HWSC0001 HWSID=HW01 RACF=Y

If you turn the RACF flag ON, ITOC will call RACROUTE REQUEST=VERIFY to verify a userid and password, the client can encrypt the password and decrypt the password in the user exit routine. HWSWEB00 provides a sample password decryption. (See "HWSWEB00 user message exit" on page 20.) The HWSWEB00 user exit was created for IMS Web to allow you flexibility to do the security checking in the user exit.



---

## Chapter 3. IMS TCP/IP OTMA Connection commands

This section describes the following commands:

- “CLOSEHWS”
- “OPENDS”
- “OPENPORT” on page 12
- “SETRACF” on page 12
- “STOPCLNT” on page 12
- “STOPDS” on page 13
- “VIEWDS” on page 14
- “VIEWHWS” on page 15
- “VIEWPORT” on page 15

All IMS TCP/IP OTMA Connection commands must be immediately preceded on the command line of the MVS system console by the reply number of the outstanding HWS reply message (for example, *nn*HWSCMD where *nn* is the reply number).

---

### CLOSEHWS

The CLOSEHWS command terminates the HWS.

#### Parameters

None

**Usage** All work that is currently in progress, or that is queued for processing, is completed before the HWS is terminated. No new work is accepted after this command has been entered and accepted.

The HWS shuts down in the following order:

1. All active units of work for TCP/IP clients/browsers are completed.
2. Communication between the HWS and IMS is terminated.
3. The HWS terminates.

#### Example

To close the HWS:

```
nnCLOSEHWS
```

---

### OPENDS

The OPENDS command starts communication between the HWS and a datastore.

#### Parameters

*datastore\_id*

Specifies the name of the datastore. This name must be defined to the HWS through the configuration member *HWSCFGxx*, and must match one of the IDs that is defined in the *DATASTORE* configuration statement or statements.

**Usage** Use this command to reestablish communication with a datastore after communication fails between the HWS and the datastore. For example, use

this command to restart communication when all activity for the datastore in the HWS is terminated, or after a STOPDS command has terminated communication with the datastore.

Use the VIEWDS command to display information about datastores if you are not sure about the activity of a particular datastore.

The OPENDS command does not affect a datastore that is already active or a datastore that is not defined to the HWS in the configuration member HWSCFGxx.

#### **Example**

To open communication to datastore IMSA:

```
nnOPENDS IMSA
```

---

## **OPENPORT**

The OPENPORT command reestablishes HWS communication with TCP/IP to allow listening on TCP/IP ports.

#### **Parameters**

*portid*

Identifies the number of the port to be opened. This port number must match one of the port numbers that is defined in the PORTID substatement of the TCPIP configuration statement in the HWSCFGxx configuration member.

**Usage** Use this command to reestablish a TCP/IP connection to allow listening on a TCP/IP port. Use this command when communication stops between the HWS and a TCP/IP port, but the HWS has not terminated.

#### **Example**

To reestablish the TCP/IP connection between the HWS and port 9999 so that the HWS can listen on that port:

```
nnOPENPORT 9999
```

---

## **SETRACF**

The SETRACF command turns on and off the RACF flag.

#### **Parameters**

*ON/OFF*

Identifies if the RACF flag is turned on or off.

**Usage** To enable or disable the RACF user identification and verification.

#### **Example**

To turn on the RACF:

```
nnSETRACF ON
```

---

## **STOPCLNT**

The STOPCLNT command immediately terminates communication with a client using a specific TCP/IP port.

#### **Parameters**

*portid*

Identifies the port that the client is using for the TCP/IP connection with the HWS. This port number must match a port number that is defined in the PORTID substatement of the TCPIP configuration statement in the HWSCFGxx configuration member.

*clientid*

Specifies the name of the client (the client name is dynamically generated by the IMS Web Runtime component).

**Usage** Work currently in progress for that client is ended.

Use this command whenever a client is unable to accept response messages being sent to it, or when a client is waiting for a nonexistent response message (for example, when an error occurred that caused a response message to be lost before it was sent back to the client).

Use the VIEWPORT command to display the name and state of the client.

**Example**

To force the HWS to terminate communication with client CLIENT01, who is communicating with the HWS using port 9999:

```
nnSTOPCLNT 9999 CLIENT01
```

---

## STOPDS

The STOPDS command immediately terminates communication between the HWS and a datastore.

**Parameters**

*datastore\_id*

Specifies the name of the datastore. This name must match an ID that is defined in a DATASTORE configuration statement of the HWSCFGxx configuration member.

**Usage** Work currently in progress for a datastore is ended and communications with that datastore and its threads are terminated. Messages that are queued for the datastore are released and the originator of the queued messages is notified. No new messages are accepted after the STOPDS command is accepted.

Use this command to release messages that are queued for an unavailable datastore or for a datastore whose queued work belongs to unavailable clients. It can also be used for any type of error situation that requires immediate termination of communication with a datastore.

Use the OPENDS command to open communication with the datastore at a later time.

**Example**

To stop communication to datastore IMSA:

```
nnSTOPDS IMSA
```

---

## STOPPORT

The STOPPORT command immediately terminates listening on a TCP/IP port.

**Parameters**

*portid*

Identifies the number of the port on which listening is to stop. This port number must match one of the port numbers that is defined in the PORTID substatement of the TCPIP configuration statement in the HWSCFGxx configuration member.

### Usage

Work currently in progress is allowed to continue for existing clients. Only the listening for new request messages on the port is terminated immediately. When existing work has completed, the port is no longer active.

Use the VIEWPORT command to display the state of the port and any clients using that port.

### Example

To stop listening on port 9999:

```
nnSTOPPORT 9999
```

---

## VIEWDS

The VIEWDS command displays the current activity of a datastore.

### Parameters

*datastore\_id*

Specifies the name of the datastore for which information is to be displayed or ALL. If a datastore name is used, this name must match the ID parameter of a DATASTORE configuration statement of the HWSCFGxx configuration file and only the information for this datastore is displayed. If ALL is used, information for all datastores that are defined in a DATASTORE configuration statement in the HWSCFGxx configuration member is displayed.

**Usage** This command displays the current information for one or all datastores. The information displayed for each datastore is:

#### Datastore Name

Name of the datastore, as defined in the ID substatement of the DATASTORE configuration statement in the HWS configuration member HCTCFGxx.

#### Status

State of the datastore, ACTIVE, NOT ACTIVE or NOT DEFINED.

**Group** XCF group name for the group to which the HWS and IMS OTMA belong.

#### Member

HWS member name in the XCF group listed.

#### Target Member

IMS OTMA member name in the XCF group listed.

### Example

To view the information for a single datastore, IMSA:

```
nnVIEWDS IMSA
```

To view the information for all datastores defined to the HWS:

```
nnVIEWDS ALL
```

---

## VIEWHWS

The VIEWHWS command displays the current activity of the HWS.

### Parameters

None.

**Usage** Information displayed includes:

#### HWS ID

Name of the HWS, as defined in the ID substatement of the HWS configuration statement in the HWSCFGxx configuration member.

#### Datastore Name

Name of the datastore, as defined in the ID substatement of the DATASTORE configuration statement in the HWSCFGxx configuration member or No active Datastores.

#### Status (Datastore)

State of the datastore, whether ACTIVE, NOT ACTIVE or NOT DEFINED.

**Group** XCF group name for the group to which the HWS and IMS OTMA belong.

#### Member

HWS member name in the XCF group listed.

#### Target Member

IMS OTMA member name in the XCF group listed.

**Port** Identifies the port or ports that are defined in the PORTID substatement of the TCPIP configuration statement in the HWSCFGxx configuration member or No active Ports.

#### Status (Port)

State of the port, whether ACTIVE or INACTIVE.

**Client** Specifies the name of the client (the client name is dynamically generated by the IMS Web Server) or NO active Clients.

#### Status (Client)

State of the client's thread, whether ACTIVE or IDLE.

You can use the VIEWDS command to display information for datastores only or the VIEWPORT command to display information for ports only.

### Example

To view the HWS:

```
nnVIEWHWS
```

---

## VIEWPORT

The VIEWPORT command displays the current activity of a port.

### Parameters

*portid*

Specifies the number of the port for which information is to be displayed or ALL. If a port number is used, this number must match a port number that is defined in the PORT substatement of the TCPIP configuration statement in the HWSCFGxx configuration member and the information is displayed for this port

only. If ALL is used, information is displayed for all ports that are defined in the TCPIP configuration statement in the HWSCFGxx configuration member.

**Usage** Information displayed includes:

**Port** Identifies the port or ports that are defined in the PORTID substatement of the TCPIP configuration statement in the HWSCFGxx configuration member.

**Status (Port)**

State of the port, whether ACTIVE or INACTIVE.

**Client** Specifies the name of the client (the client name is dynamically generated by the IMS Web Runtime component).

**Status (Client)**

State of the client's thread, whether ACTIVE or IDLE.

**Example**

To view the information for a single port, 9999:

```
nnVIEWPORT 9999
```

To view the information for all ports defined to the HWS:

```
nnVIEWPORT ALL
```

---

## Chapter 4. IMS client for Java

This section describes the contents of JAVASAMP.exe and how to run the sample Java client. Refer to the *IMS TCP/IP OTMA Connection User's Guide* for downloading and installation instructions, as well as how to modify the Java client.

---

### Contents of JAVASAMP.exe

JAVASAMP.exe is a self-extracting, compressed file (see the attention statement in the next paragraph) that contains a sample Java client along with documentation. JAVASAMP.exe is packaged with IMS TCP/IP OTMA Connection (ITOC) in HWSMH $v$ rm.exe, where  $v$ rm represents the level (version, release, modification) of IMS TCP/IP OTMA Connection. This sample client for Java consists of code used to prepare a Java program that accesses IMS applications and data.

**Attention:** ITOC and the IMS Client for Java sample program are packaged using Info-ZIP's compression utility to create a self-extracting executable archive (zip) file. This program uses UnZip internally to extract the archived files. Info-ZIP's software (Zip, UnZip and related utilities) is free and can be obtained as source code or executables from various anonymous-ftp sites, including <ftp.uu.net:/pub/archiving/zip/> or from the Web at <http://www.cdrom.com/pub/infozip/>.

**Requirement:** Both IMS Version 5 OTMA and the IMS TCP/IP OTMA Connection are prerequisites to using the sample.

The client source code is written in Java and includes the following files:

- ClientLauncher.java
- ClientLauncherApplet.java
- ClientLauncherFrame.java
- FrameInput.java
- CanvasAbout.java
- IMSCanvas.java
- SampleTran.java
- AboutDialog.java
- QuitDialog.java

The following files are also contained in the JAVASAMP file:

- ClientLauncher.html
- ims2.gif (used by the applet)

The documentation files are:

- JavaSampleChanges.text
- JavaSampleChanges.html
- JavaSampleInstallation.text
- JavaSampleInstallation.html

---

## Running the sample Java client

You can use the sample Java program as a Java applet or a Java application, depending on how you start it. The overhead of implementing Java clients in this way is minor, compared with the benefits.

In order for Java to run correctly on NT when a Netscape browser is used, you must define a user variable in your NT environment, as follows:

```
CLASSPATH=.;
```

## Running the client as a Java application

In order to run the sample Java client as an application, you must have installed the Java Development Kit on your workstation. After the sample Java source code has been compiled, enter `java ClientLauncher` at a command line prompt to run the sample client as a Java application.

## Running a client applet locally

Open `ClientLauncher.html` as a local file from a Java-enabled Web browser that supports JDK 1.1.3 and an applet for socket codes.

## Running the client as a remote applet

Before you can run the sample Java client as a remote applet, your Internet server must be running on the host where the IMS TCP/IP OTMA Connection is running (must have the same MVS image). Copy all class, .gif, and .html files to your MVS Internet Connection Server html directory in the same MVS image where the IMS TCP/IP OTMA Connection is running, and then bring up the `ClientLauncher` (`ClientLauncher.html`) in a Java-enabled Web browser.

`ClientLauncher` starts a new Transaction Input client frame every time you click the Start A Client button. If you start more than one client windows, the newer windows hide the existing client window. All of these client frame windows run independently, so you must close each window individually if you are running the sample Java client as an applet. If you are running the sample client as a Java application, you can either close the client frame windows individually (as you must do if the sample Java client is running as an applet) or all at the same time by closing the original window (the `ClientLauncher` running as an application). When you close the `ClientLauncher` application, all client frame windows that were generated by that application close automatically.

---

## Chapter 5. User message exits for ITOC

Three message exits are provided with ITOC. Each of the message exits allows you to call IMSSEC, the security message exit, issue the RACF function in these user message exit routines, or use the ITOC user RACF function.

---

### EZAIMSO0 user message exit

EZAIMSO0 is shipped as object code only (OCO) with ITOC. The message exit is shipped as OCO to allow ITOC and the user message exit provided by TCP/IP to be able to stay in sync without requiring an upgrade of both products at the same time when message exit functions change. If the installation uses this message exit as its default, you must concatenate the RESLIB that contains the ITOC-supplied message exit in front of the TCP/IP RESLIB to ensure the correct message exit is used.

**Recommendation:** If the EZAIMSO0 user message exit does not provide the installation options, modify and use HWSSMPL0. See the *IMS TCP/IP OTMA Connection User's Guide* for information about installing this message exit.

The EZAIMSO0 user message exit provides the following functions:

- Performs data translation of ASCII to EBCDIC for input messages.
- Performs data translation of EBCDIC to ASCII for output messages.
- Builds the ITOC message structure (BPE and OTMA headers).
- If IMSSEC (the security message exit) is link-edited with this message exit then the security message exit is called.
- Defaults to COMMIT MODE=1.
- Defaults to SYNC LEVEL=NONE.
- Sets up RACF options.
- Analyzes the following message header options:
  - COMMIT MODE to override default.
  - SYNC LEVEL to override default.
  - MFS MOD name.
  - ACK/NACK/DEALLOCATE.
  - RACF options.
  - If no Client ID is passed to the exit, the message exit will generate the Client ID.

**Restriction:** Do not use the TCP/IP supplied message exit (EZAIMSO0) for ITOC 2.1. This message exit does not support ITOC 2.1.

If any errors occur with the TCP/IP translate table, report those problems to TCP/IP and notify them of the TCP/IP release you are currently using. When corrections have been made to the translate table, simply relink-edit this exit to pick up the corrected translate tables. If any errors occur with the ITOC user message exit (EZAIMSO0) code, report those problems to ITOC for corrections.

---

## HWSSWEB00 user message exit

HWSSWEB00 and the related MACROS are shipped as source code. The reason this message exit is shipped as source code is to allow you the ability to modify the message exit for installation uniqueness.

HWSSWEB00 provides two password decryption methods:

- Text string decryption
- Binary number decryption

If you choose to use the IMS Web Runtime default encryption method, you must enable one of the decryption methods in this exit. The default for the exit is password decryption disabled. You can also replace the default encryption method with your own method.

**Requirement:** HWSSWEB00 is the user message exit for IMS Web. If you change the IMS Web workstation security, be sure and change this message exit too.

---

## HWSSMPL0 user message exit

HWSSMPL0 and the related MACROS are shipped as source code. The reason this message exit is shipped as source code is to allow you the ability to modify the message exit for installation uniqueness.

See the *IMS TCP/IP OTMA Connection User's Guide* for information about installing this message exit.

The HWSSMPL0 user message exit provides the following functions:

- Performs data translation of ASCII to EBCDIC for input messages.
- Performs data translation of EBCDIC to ASCII for output messages.
- Builds the ITOC message structure (BPE and OTMA headers).
- If IMSSEC (the security message exit) is link-edited with this message exit then the security message exit is called.
- Defaults to COMMIT MODE=1.
- Defaults to SYNC LEVEL=NONE.
- Sets up RACF options.
- Analyzes the following message header options:
  - COMMIT MODE to override default.
  - SYNC LEVEL to override default.
  - MFS MOD name.
  - ACK/NACK/DEALLOCATE.
  - RACF options.
  - If no Client ID is passed to the exit, the message exit will generate the Client ID.

---

# Readers' Comments — We'd Like to Hear from You

IMS TCP/IP OTMA Connection Programmer's Reference

Publication No. itocpr-0002-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



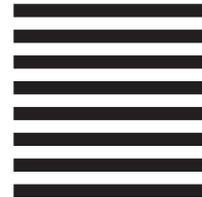
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Department BWE/H3  
P.O. Box 49023  
San Jose, CA 95161-9945



Fold and Tape

Please do not staple

Fold and Tape





Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.