Security Options and Considerations for:

IMS/Open Transaction Manager Access (OTMA), IMS Connect, and the MQSeries-IMS Bridge Application

May 20, 2002

Author: Alonia (Lonnie) Coleman, Dallas Systems Center (DSC)

Technical Editors: Suzie Wendler (DSC)
Jack Yuan, Silicon Valley Laboratory (SVL)

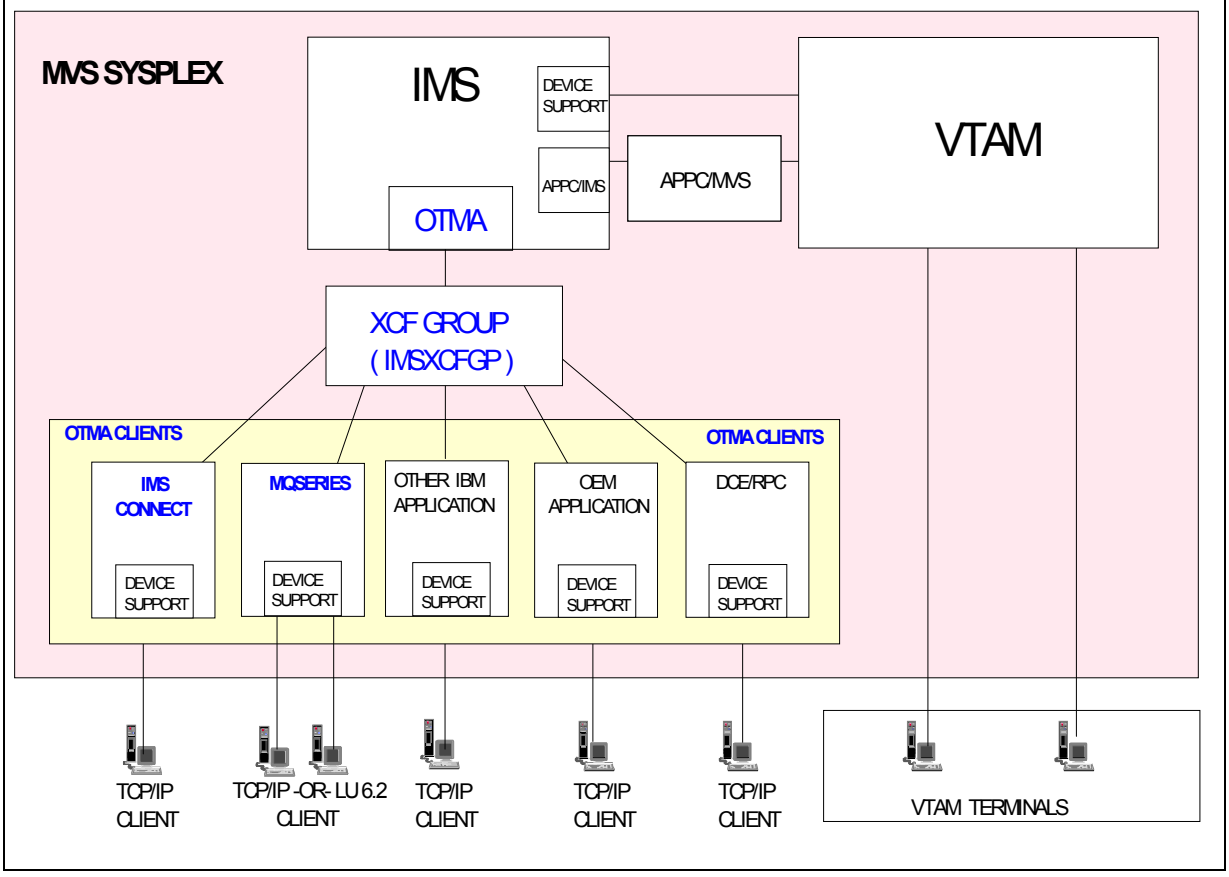
Other Contributors: Gerald Hughes, SVL
Barbara Klein, SVL
Geoff Nicholls, IBM Australia

Introduction

Traditionally, IMS transactions have been entered at a terminal on an SNA network. The transactions were sent to IMS via VTAM. Today other types of networks are supported, the most notable being TCP/IP. The Open Transaction Manager Access (OTMA) feature of the IMS Transaction Manager (IMS TM) allows you to use IMS TM as a server to many different MVS applications that participate in client-server environments.

OTMA provides a high performance, client-server protocol for IMS to communicate very efficiently with non-VTAM applications such as TCP/IP, IMS Connect, and MQSeries for z/OS. OTMA communicates with these MVS applications using MVS's high-performance cross-system coupling facility (XCF) function. The MVS applications that access IMS through OTMA are called 'OTMA clients'. IMS/OTMA and the OTMA client applications compose an XCF group where OTMA and MVS 'OTMA client' applications are XCF group members. See Figure 1, 'Using XCF to Connect Networks', below.

Figure 1. Using XCF To Connect Networks



As companies utilize OTMA to provide access to IMS by TCP/IP users, securing IMS resources has become more critical. Therefore, it is important for an installation's security staff to have a good understanding of how to restrict access to IMS resources from traditional sources as well as from other sources such as TCP/IP networks.

Several IBM products use the OTMA interface to communicate with IMS. One of the IBM products that provides access to IMS resources from TCP/IP networks is IMS Connect. As companies expand their use of IMS/OTMA and IMS Connect, it has become apparent that new security related ***and*** non-security related functional enhancements are needed for IMS/OTMA and IMS Connect users. IBM's IMS/OTMA and IMS Connect development staffs have plans to provide the functional enhancements that are required as soon as possible.

This white paper addresses several customer requests, including:

- A technical overview of IMS/OTMA security options. The following are described:
  - OTMA security for client-bid requests as well as security for IMS commands and IMS transactions received via OTMA when the OTMA security level has been defined as NONE, CHECK, FULL, or PROFILE.
  - The use of optional user security exit routines for authorizing access to IMS commands and/or IMS transactions received via OTMA, along with a description of when these exit routines are invoked and when they are used in conjunction with RACF. The applicable exit routines include the: Command Authorization Exit (DFSCCMD0), Transaction Authorization Exit (DFSCTRN0), and the Security Reverification Exit (DFSCTSE0).

    Also, describe when user security exit routines are invoked when used in conjunction with RACF.
  - An update on the status of security-related enhancements that are planned.
- A description of the security capabilities of IMS Connect along with an update on the status of security-related enhancements that are planned for IMS Connect.
- A technical overview of MQSeries-IMS Bridge security .

Basic Concepts and Terminology

An understanding of the terminology used throughout this document is helpful in understanding IMS/OTMA security. Some of the terminology used includes:

- IMS resource
- OTMA client
- Client-bid
- Security facility
  - Resource Access Control Facility (RACF)
  - User exit routine
- OTMA security level.

End users enter messages that are sent to IMS for processing. The messages contain requests for IMS resources such as IMS commands or transaction codes. Transaction codes are processed by application programs that access data stored in IMS database data sets. During the normal operation of an IMS system many requests are received for access to IMS resources. **IMS resources** include the following:

- Transaction codes
- Commands
- Terminals
- Program specification blocks (PSBs)
- Databases
- Data sets
- Dependent regions
- Control region.

Requests for access to IMS resources may originate from many sources such as OTMA, APPC, an IMS static or ETO terminal, or an application program running in an IMS dependent region. The OTMA interface is used by **OTMA clients** to send input messages to IMS and receive output response messages from IMS. **OTMA clients** are OS/390 and z/OS programs, such as IMS Connect and the MQSeries Bridge Application, that use the OTMA interface to send/receive messages to/from IMS.

Before an OTMA client such as IMS Connect can pass IMS transaction or IMS command messages entered by end users to IMS using the OTMA interface, the OTMA client must first join the same XCF group as the IMS to which it will transmit messages. Figure 1, *'Using XCF to Connect Networks'* illustrates this point. When both the OTMA client and IMS have successfully joined the same XCF group, the OTMA client must issue a *request to connect* to IMS/OTMA. Two subsystems can not communicate with each other merely by joining the same XCF group. To illustrate this point, suppose the subsystem labeled 'OEM Application' in Figure 1 intends to communicate with the subsystem labeled 'Other IBM Application' rather than with IMS/OTMA. Therefore, not only must subsystem members of an XCF group join the same group, a connection must be established between the subsystems in order for them to communicate. The ***request to connect*** to a specific IMS subsystem using the OTMA interface is called a **client-bid**. In addition to being a connection request process, a *'client-bid'* is also a special type of message sent to IMS/OTMA during the client-bid process. The client-bid message must be the ***first*** message sent by the OTMA client to IMS/OTMA. **All messages** destined for IMS/OTMA (including **client-bid messages** as well as subsequent **end user messages** that request the execution of IMS commands and transaction codes) contain security-related information. The security-related information consists of one or more of the following fields in an OTMA prefix:

- The **userid** of the OTMA client, such as IMS Connect subsystem userid, when the message is a client-bid message or the **userid** of an end user when the message originated from end user and requests the execution of an IMS command or transaction code.

- A **UTOKEN** which is an acronym for 'user token'. The presence of a UTOKEN in the security-related fields in an OTMA prefix indicate that an authorized subsystem has previously invoked RACF (or an equivalent security product) to validate the userid in the message prior to sending the message to IMS/OTMA.

  When **RACF security checking has been activated** (turned on) **in IMS/OTMA** for messages received via OTMA clients, each client-bid message must contain a valid UTOKEN or the client-bid (request to connect) **will fail**. The presence of a

valid UTOKEN in the client-bid message indicates to IMS/OTMA that the OTMA client has the authority to act as an OTMA client.

Each end user message may or may not contain a UTOKEN field in the security-related information in an OTMA prefix. If an OTMA client such as IMS Connect invoked RACF (or an equivalent) to validate the userid in the message the UTOKEN is passed to IMS/OTMA in an OTMA prefix. If the OTMA client did not invoke RACF to validate the userid in the message the userid (and optionally, a profile name) is passed to IMS/OTMA in an OTMA prefix.

In any event, if a message contains both a userid and a UTOKEN, **IMS/OTMA uses the UTOKEN for userid validation instead of the userid.** From a performance perspective use of the UTOKEN is more efficient when RACF is invoked to validate a userid.

- A **profile name** (also referred to as the **RACF group**) of which the OTMA client or the end user's userid is a member.

- A **1-byte security flag** field that may contain only **one** of three possible values: **N**, **C**, or **F**. The values (N for NONE, C for CHECK, and F for FULL) correspond to values supplied by the OTMASE= startup parameter and the /SECURE OTMA command. It should be noted that IMS only uses (or honors) the value in the security flag field with the OTMA security level PROFILE. When any of the other OTMA security level are used (namely NONE, CHECK, or FULL) IMS does **not** use the security flag field value to determine whether or not to invoke RACF.

  The value specified in the security flag field indicates whether or not RACF should be called to perform the following types of authorization checking:

  - Authorize the client-bid (connection request) if the message is a client-bid message.
  - Verify the end user's userid in a message if the message is one that requests the execution of an IMS command or transaction code.
  - Perform resource authorization checking to determine if the userid in the message is authorized to access the IMS command or transaction code requested.

As OTMA clients send messages (whether a client-bid message or a message from an end user requesting the execution on an IMS command or transaction) to IMS/OTMA, IMS/OTMA must decide on an action to take with respect to security. IMS/OTMA must decide to either:

- Bypass security checking for the incoming message and accept it for processing without invoking a security facility.

- Invoke one or more of the following **security facilities** (a security product such as RACF and/or user exits) to perform authorization checking:

  - The **Resource Access Control Facility** (RACF) which is a program product that runs on OS/390 and z/OS operating systems and provides resource authorization checking and userid validation functions.

  - **Optional** **user exit routines** which may be provided by the customer to perform authorization checking functions. IMS provides three sample user exit routines that the customer may modify to perform authorization checking in IMS systems configured to use OTMA.

    These user exit routines are:

    The **Command Authorization Exit Routine** (DFSCCMD0) which is invoked to provide authorization checking for OTMA messages that request the execution of an IMS command.

    The **Transaction Authorization Exit Routine** (DFSCTRN0) which may be invoked to provide authorization checking for OTMA messages that request the execution of an IMS transaction code. This exit may also be invoked during transaction processing if the application program issues one or more of the following DL/I calls: CHNG AUTH, and/or ISRT.

The **Security Reverification Exit Routine** (DFSCTSE0) which is invoked to provide authorization checking when an application that is processing an OTMA message issues a CHNG call and/or an AUTH call.

Although RACF security checking and user exit routine security checking are optional and do not have to be included in the IMS/OTMA system, it is important to understand the conditions under which IMS/OTMA either invokes or does not RACF and/or the user exit routines.

IMS/OTMA determines whether or not to invoke RACF for messages received via OTMA clients based on the **OTMA security level** assigned to the IMS system. Either an IMS startup parameter, OTMASE=, or the /SECURE OTMA command may be used to establish the OTMA security level. Additional information is provided on OTMA security levels in the sections 'Technical Overview of IMS/OTMA Security Options' and ' OTMA Security Levels'.

Technical Overview of IMS/OTMA Security Options

IMS provides the following ***optional*** RACF security checking in OTMA environments:

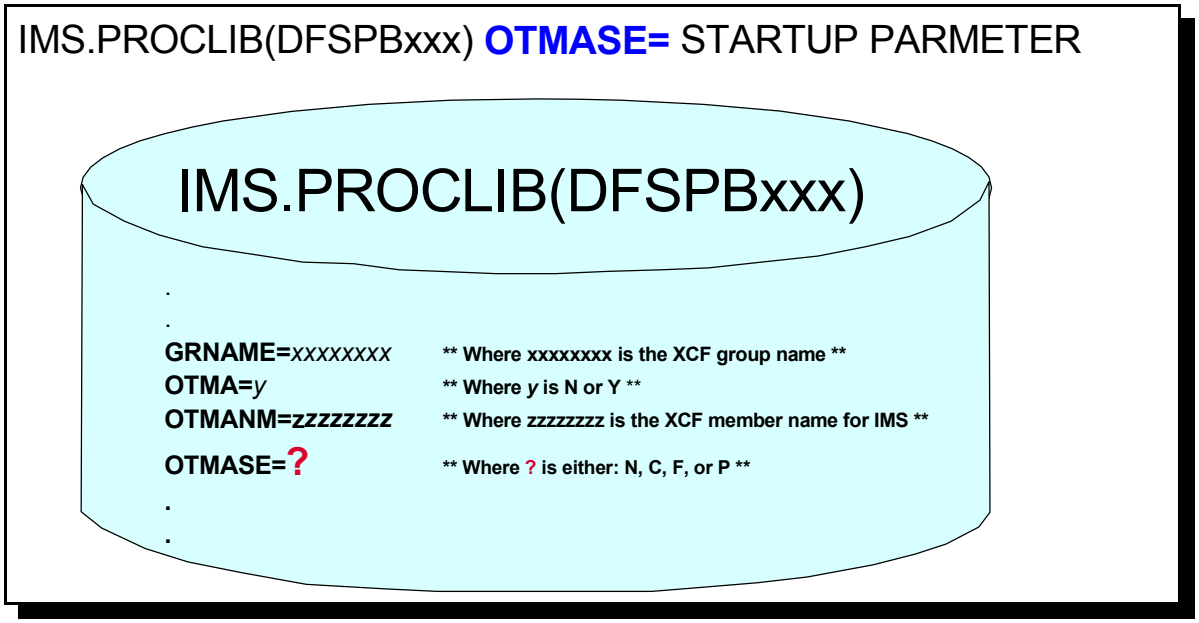
- **Client-bids** which determine whether OTMA clients such as IMS Connect and MQSeries Bridge can connect to IMS for the purpose of sending end user messages to IMS for processing.
- **IMS commands** entered by end users attached to OTMA clients.
- **IMS transactions** entered by end users attached to OTMA clients

The **OTMA *'security level'*** for an IMS system determines whether or not IMS ***calls RACF*** to perform authorization checking for any of the above activities. It should be noted, however, regardless of any OTMA security level, IMS will always invoke certain security exits if they exist. These include the Command Authorization Exit and the Security Reverification Exit. IMS invokes the Transaction Authorization Exit based on the following:

- The OTMA security level specified for the IMS system. If the Transaction Authorization Exit has been included in the IMS system it is always invoked by IMS when the OTMA security level is NONE.
- Whether RACF was invoked to perform transaction authorization processing and the RACF resulting return code. Other OTMA security levels, namely CHECK and FULL, result in IMS invoking RACF for transaction authorization processing. If the Transaction Authorization Exit has been included in the IMS system it is invoked if and only if RACF does not deny authorization. In other words, if RACF grants a userid authorization to a transaction or if the transaction is not secured by RACF the exit routine is invoked, whereas if RACF denies authorization to the transaction the exit routine is not invoked.

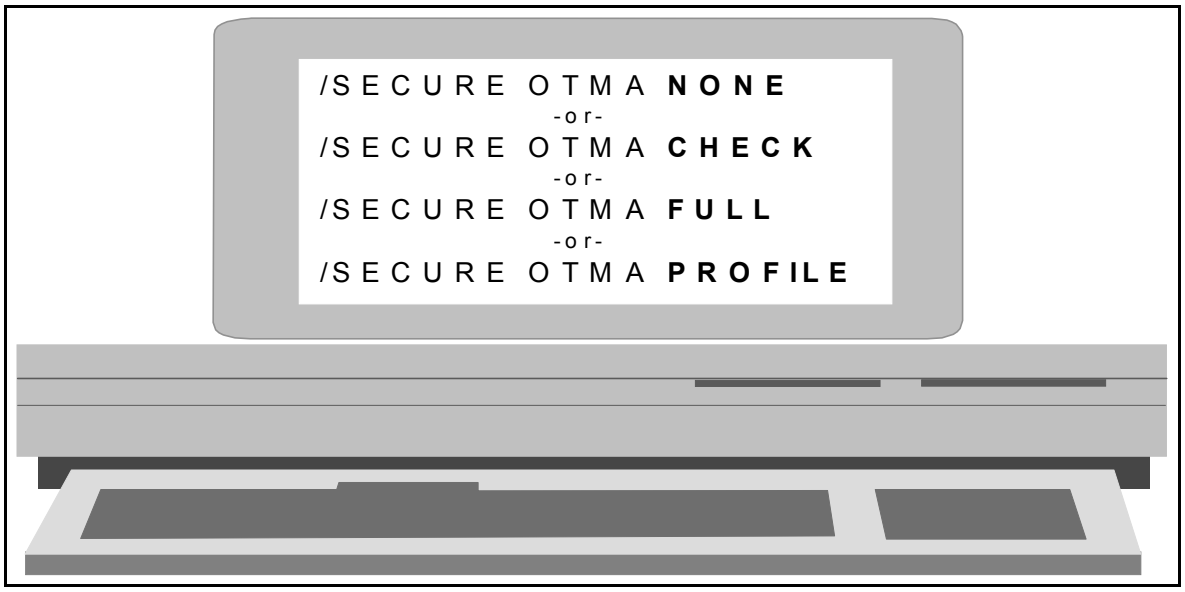
OTMA Security Levels

There are four OTMA security levels: NONE, CHECK, FULL, and PROFILE. Each customer must decide which ***one*** of the OTMA security levels best meets the installation's requirements. The OTMA security level for an IMS system may be established by an IMS startup parameter specification, **OTMASE=,** in the DFSPBxxx member of the IMS procedure library or by the **/SECURE OTMA** command.



By default, the OTMA security level in IMS is set to **FULL** (or F). To override the default, specify a different value for OTMASE. The valid values for the OTMASE= parameter are: **N** (for NONE), **C** (for CHECK), **F** (the default, for FULL), and **P** (for PROFILE).

The /SECURE OTMA command may be used to specify or to change the OTMA security level after IMS has been started. The OTMA security level may be changed by issuing ***one*** of the following keywords on the /SECURE OTMA command: **NONE**, **CHECK**, **FULL**, or **PROFILE**. The following diagram illustrates use of the /SECURE OTMA command.



The OTMASE= startup parameter specification and the /SECURE OTMA command do the same thing. Each establishes the OTMA security level for an IMS system. The **/SECURE OTMA** command is provided to allow you to override the OTMA security level set by the **OTMASE=** parameter specification used during IMS initialization. Use of the /SECURE OTMA command allows the OTMA security level to be changed without having to reinitialize IMS.

Although the /SECURE OTMA command overrides the value specified by OTMASE=, the OTMA security level specified on the /SECURE OTMA command is ***not*** maintained across an IMS restart. When IMS is restarted, the OTMA security level will be established by the:

- Value specified on the OTMASE= keyword, or
- Default value, OTMASE=F. This applies when the OTMASE= keyword is not coded in startup parameters or when the OTMASE= keyword is specified in the startup parameters without a value.

The table below shows how to set each of the OTMA security levels.

OTMA SECURITY LEVEL	SECURITY LEVEL SET BY EITHER:	
	STARTUP PARAMETER	EQIVALENT COMMAND
NONE	OTMASE=**N** OTMASE=**P +** security flag value '**N**'	/SEC OTMA **NONE** /SEC OTMA **PROFILE +** security flag value '**N**'
CHECK	OTMASE=**C** OTMASE=**P +** security flag value '**C**'	/SEC OTMA CHECK /SEC OTMA **PROFILE** + security flag value '**C**'
FULL	OTMASE=**F** OTMASE=**P +** security flag value '**F**'	/SEC OTMA **FULL** /SEC OTMA **PROFILE** + security flag value '**F**'
PROFILE	OTMASE=P	/SEC OTMA PROFILE

NOTE: WHEN PROFILE (OR 'P') IS USED THE SECURITY FLAG VALUE IN EACH MESSAGE RECEIVED VIA OTMA IS CHECKED TO DETERMINE WHETHER AN OTMA LEVEL OF NONE, CHECK, OR FULL SHOULD BE USED FOR THAT MESSAGE.

Invoking Security Facilities

From an IMS perspective once a message has been received by OTMA, the security checking is business as usual. When OTMA security is activated, upon receipt of a message IMS issues invokes RACF to perform one of the following:

- If the incoming message contains a valid UTOKEN, build an ACEE from the information in the UTOKEN and return the ACEE to IMS.
- If the incoming message does not contain a UTOKEN, validate that the userid in the incoming message has been defined to RACF and build (and return to IMS) an ACEE for valid userids.

The reason IMS needs to obtain an ACEE (RACF security control block) is so that subsequent RACF security checking can take place for that userid. For example, an ACEE is needed for an OTMA client before RACF client-bid security checking can take place. Likewise an ACEE is needed for an end user before RACF security checking take place to determine if the userid for the end user is authorized to execute and IMS command or transaction.

IMS/OTMA determines which, if any, security facilities need to be invoked for messages received via OTMA based on the OTMA security level and on whether or not security exits have been included in the system. A thorough understanding of the impact and considerations for use of each of the OTMA security levels is required in order to select the OTMA security level that best meets each customer's needs. It is important for customers to weigh the pros and cons for use of each of the OTMA security levels. Each OTMA security level is described in greater detail in subsequent topics.

OTMASE=N or /SECURE OTMA NONE

When the OTMA security level is **NONE**, RACF is **not** invoked by IMS. **OTMASE=N** and **/SECURE OTMA NONE** establish an IMS-wide security level. An IMS-wide level indicates that IMS takes the same action for each message received via OTMA. For an OTMA security level of NONE, this means that the IMS subsystem will **not** invoke RACF to perform**:**

- Client-bid security checking for client-bid messages received.
- IMS command authorization for command messages received via OTMA.
- IMS transaction authorization for initial input messages received via OTMA.

The Client-bid

The client-bid message may contain one or more of the following security fields:

- A **UTOKEN** which must be included in the client-bid message so that IMS/OTMA can validate the OTMA client's authority to act as an OTMA client.
- The **userid** of the OTMA client, such as IMS Connect subsystem userid.
- A **profile name** (also referred to as the **RACF group)** of which the OTMA client is a member.
- A **1-byte security flag** which indicates whether RACF should be called to authorize the client-bid (connection request).

Client-bid Security Checking For Security Level NONE

Client-bid security checking does not take place when the OTMA security level is **NONE**. This results in the success of all client-bid requests to that particular IMS. Therefore OTMA clients like IMS Connect and the MQSeries Bridge Application will automatically be allowed to connect to IMS.

All of the security fields (userid, profile, security flag, and/or UTOKEN) in client-bid messages are ***ignored*** when the OTMA security level is NONE.

After receipt of the client-bid message, IMS/OTMA sends an ACK (acknowledgment) message to the OTMA client to indicate a successful client-bid. After receiving the ACK, the OTMA client can start sending TCP/IP end user messages (which contain requests to execute IMS commands and IMS transactions) to IMS for processing.

IMS Command Authorization For Security Level NONE

RACF is not called for command authorization when the OTMA security level is NONE. If the installation does ***not*** use the Command Authorization Exit Routine (DFSCCMD0) to authorize users to execute IMS commands, IMS automatically provides '**default command security'** when the OTMA security level is NONE. Default command security restricts the commands received via OTMA clients to the following commands: **/BROADCAST**, **/LOCK**, **/LOG**, **/RDISPLAY**, and **/UNLOCK**.

To deactivate (turn off) 'default command security' and allow additional IMS commands received via OTMA to be executed when the OTMA security level is **NONE** the Command Authorization Exit Routine (DFSCCMD0) must be customized and included in the IMS system. The Command Authorization Exit Routine (DFSCCMD0) is coded to verify that the end user's userid in the incoming OTMA message is authorized to execute the IMS command specified in the message. The sample Command Authorization Exit (DFSCCMD0) that is provided with the IMS system allows the same five (5) default commands as does 'default command security' for commands received via OTMA. Therefore the sample exit must be modified to allow additional IMS commands received via OTMA clients to be executed.

IMS Transaction Authorization For Security Level NONE

As with command authorization, when the OTMA security level is **NONE,** RACF is not called. However, if you have included the Transaction Authorization Exit Routine (DFSCTRN0) in the IMS system, it ***will always be invoked*** to perform transaction authorization checking for transactions received via OTMA when the OTMA security level is **NON**E. The Transaction

Authorization Exit is invoked before IMS places the incoming OTMA message on the message queue. This exit routine is also invoked for other resources (such as DATABASEs, SEGMENTs, FIELDs, and/or OTHER resources) requested by the application program on an AUTH call and for a transaction code set as the destination on a CHNG call .

Transaction Authorization Exit Routine

The Transaction Authorization Exit Routine (DFSCTRN0) is coded to verify whether the userid in an incoming message is authorized to execute the transaction code specified in the message.

Upon entry to the exit, IMS passes DFSCTRN0 a parameter list. For transaction messages received via OTMA, the following is some of the information passed to the exit:

- A pointer to the userid.
- A pointer to the transaction pipe (TPIPE) name used to transmit the transaction message to IMS.
- A pointer to the XCF member name of the OTMA client.
- A pointer to the transaction code, database, segment, field or other IMS resource that is requested.
- Address of the System Content Directory (SCD).

DFSCTRN0 may be customized to use the above information in transaction authorization processing for input transaction messages.

Upon return to IMS from DFSCTRN0, Register 15 (R15) is set to one of the following return codes:

- **0** - RC=0 indicates that IMS should accept the transaction. When DFSCTRN0 is coded to allow the userid in the incoming OTMA message to execute the transaction, IMS places the transaction message on the message queue and schedules the transaction for processing.

- **4** - RC=4 indicates that the transaction is not protected. By default, IMS allows access to unprotected transactions and takes the same action as for RC=0.

- **8** - RC=8 indicates that the userid in the incoming OTMA message is not authorized to the transaction. When DFSCTRN0 denies access, IMS does not queue the message to the message queue. Instead, IMS rejects the message and sends a DFS1292E security violation message to the OTMA client. The OTMA client sends the message to the end user.

If the Transaction Authorization Exit allows the userid in the incoming OTMA message to execute the transaction, the message is queued to the message queue and an application program is scheduled to process the transaction. The application issues a Get Unique (GU) call to retrieve the message from the message queue for processing. During transaction processing, the application program may do one or more of the following:

- Issue a CHNG call with the destination set to a transaction code.
- Issue an AUTH call with the CLASSNAME specified as TRAN, DATABASE, SEGMENT, FIELD, or OTHER.
- Perform a deferred conversational program-to-program message switch . A deferred conversational program-to-program message switch is an ISRT call that specifies a SPA for a conversational transaction.

In the above cases, the Transaction Authorization Exit Routine (DFSCTRN0) is invoked to determine whether the userid in the original input message is authorized to one of the following resources:

- The transaction code requested via the CHNG call.
- The transaction code, database, segment, field, or other resource requested via the AUTH call.
- The transaction code requested via the ISRT call for a deferred conversational program-to-program message switch.

The exit sets a return code of either 0, 4, or 8 in R15 to indicate whether access to the resource should be granted or denied.

Security Reverification Exit Routine

The Security Reverification Exit Routine (DFSCTSE0) is a ***special entry point*** in the Transaction Authorization Exit Routine (DFSCTRN0). It is used to provide authorization processing for transactions requested via DL/I CHNG and for database, segment, field, and other resources requested via the DL/I AUTH call.

If the Security Reverification Exit Routine (DFSCTSE0) is included in the IMS system, it will be invoked ***after*** the Transaction Authorization Exit (DFSCTRN0). The difference between this exit and DFSCTRN0 is that the Security Reverification Exit Routine (DFSCTSE0) is ***only*** invoked to verify a userid's authority to access a transaction, database, segment, field, or other resource requested by a **CHNG** or an **AUTH** call. The Transaction Authorization Exit (DFSCTRN0) is invoked for source **input** transaction messages ***as well as*** for transactions, databases, segments, fields, and other resources requested by CHNG calls, AUTH calls, and deferred conversational program-to-program message switches.

The Security Reverification Exit Routine (DFSCTSE0) is passed the return code from the Transaction Authorization Exit (DFSCTRN0). This exit makes the ***final decision*** on whether the userid in the initial OTMA transaction message is allowed to access the transaction, database, segment, field, or other resource requested via the CHNG or AUTH call, regardless of the return code from the Transaction Authorization Exit (DFSCTRN0).

Considerations For Using OTMA Security Level NONE

There are several factors that you should consider prior to using an OTMA security level of NONE. These include:

- Some customers perform security checking on the TCP/IP client platform and/or on the OTMA client running on z/OS. These customers may not desire any additional security checking to be performed by RACF on behalf of IMS/OTMA. In these situations, an OTMA security level of **NONE** is acceptable because it allows the installation to bypass security for messages received via OTMA.

- It ***is possible*** for RACF to be invoked for resource authorization processing even when you have specified an OTMA security level of **NONE**! Although RACF is not called for the source input message received via OTMA, RACF may be invoked if the application which processes the source input message requests one of the following:

  a. A different transaction code via a CHNG or AUTH call during transaction processing.
  b. A database, segment, field, or other data resource via an AUTH call.
  c. A transaction code via an ISRT call for a deferred conversational program-to-program message switch.

  If IMS is tailored to invoke RACF for transaction authorization for non-OTMA entered transactions, then in the above instances RACF is invoked for authorization checking for resources requested via CHNG calls, AUTH calls, and deferred conversational program-to-program message switches. Maintenance must be applied to the IMS system to cause IMS to bypass invoking RACF for resources requested via these application-issued calls.

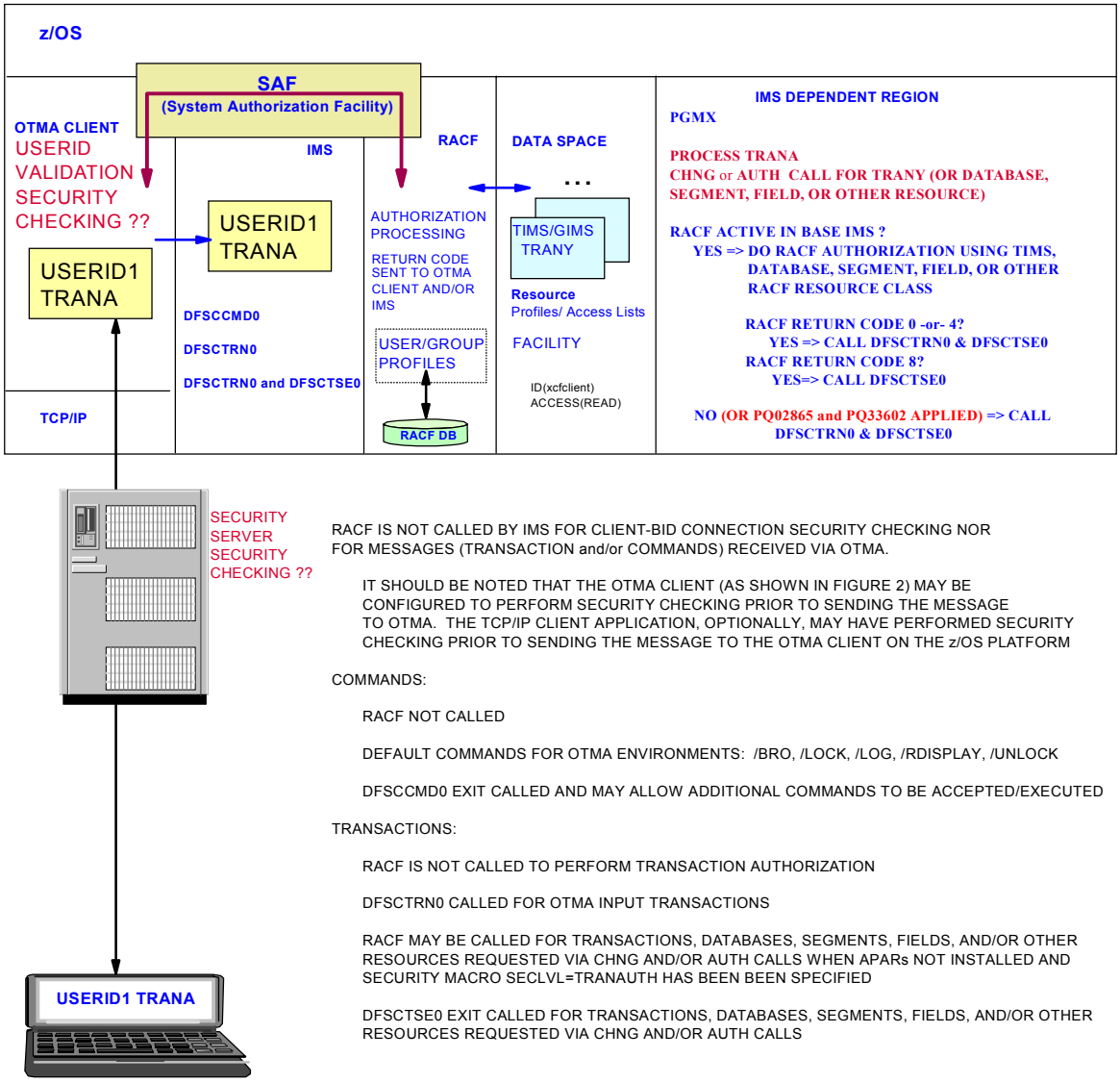
  Two PTFs must be applied to bypass resource (where a resource is transaction, database, segment, field, and other resource) authorization checking altogether when the OTMA security level is NONE. The maintenance is listed below by IMS version number.

  - IMS Version 6
    - APAR PQ02865 - PTF UQ05169
    - PQ33602 - PTF UQ41660
  - IMS Version 7
    - APAR PQ33603 - PTF UQ41663

An OTMA security level of NONE really means NONE when the maintenance has applied to IMS. The choice is yours. If you really do want to verify the TCP/IP end user's authority to access a transaction, database, segment, field, or other resource requested by a CHNG call, an AUTH call, or a deferred conversational program-to-program message switch then do not apply the maintenance cited above.

Figure 2, *'/SECURE OTMA NONE or OTMASE=N '* summarizes the concepts associated with an OTMA security level of NONE.

FIGURE 2. /SECURE OTMA NONE or OTMASE=N



RACF IS NOT CALLED BY IMS FOR CLIENT-BID CONNECTION SECURITY CHECKING NOR FOR MESSAGES (TRANSACTION and/or COMMANDS) RECEIVED VIA OTMA.

IT SHOULD BE NOTED THAT THE OTMA CLIENT (AS SHOWN IN FIGURE 2) MAY BE CONFIGURED TO PERFORM SECURITY CHECKING PRIOR TO SENDING THE MESSAGE TO OTMA. THE TCP/IP CLIENT APPLICATION, OPTIONALLY, MAY HAVE PERFORMED SECURITY CHECKING PRIOR TO SENDING THE MESSAGE TO THE OTMA CLIENT ON THE z/OS PLATFORM

COMMANDS:

RACF NOT CALLED

DEFAULT COMMANDS FOR OTMA ENVIRONMENTS: /BRO, /LOCK, /LOG, /RDISPLAY, /UNLOCK

DFSCCMD0 EXIT CALLED AND MAY ALLOW ADDITIONAL COMMANDS TO BE ACCEPTED/EXECUTED

TRANSACTIONS:

RACF IS NOT CALLED TO PERFORM TRANSACTION AUTHORIZATION

DFSCTRN0 CALLED FOR OTMA INPUT TRANSACTIONS

RACF MAY BE CALLED FOR TRANSACTIONS, DATABASES, SEGMENTS, FIELDS, AND/OR OTHER RESOURCES REQUESTED VIA CHNG AND/OR AUTH CALLS WHEN APARs NOT INSTALLED AND SECURITY MACRO SECLVL=TRANAUTH HAS BEEN BEEN SPECIFIED

DFSCTSE0 EXIT CALLED FOR TRANSACTIONS, DATABASES, SEGMENTS, FIELDS, AND/OR OTHER RESOURCES REQUESTED VIA CHNG AND/OR AUTH CALLS

OTMA Security Level NONE Summary

The following table summarizes the security actions taken by IMS/OTMA when the security level is **NONE**.

OTMA SECURITY LEVEL **NONE**	MESSAGE ORIGIN			TRAN CODE REQUESTED VIA CHNG CALL	DATABASE, SEGMENT, FIELD, OTHER, OR TRAN CODE REQUESTED VIA AUTH CALL	CONVERSATIONAL TRAN CODE REQUESTED VIA ISRT CALL (DEFERRED CONVERSATIONAL PROGRAM SWITCH)
	OTMA CLIENT	END USER				
	CLIENT-BID	IMS COMMAND	IMS TRANSACTION			
RACF	NOT INVOKED ALL CLIENT-BIDS ARE ALLOWED	NOT INVOKED 5 DEFAULT OTMA COMMANDS ALLOWED	NOT INVOKED	**INVOKED** IF APAR IS NOT APPLIED	**INVOKED** IF APAPAR IS NOT APPLIED	**INVOKED** IF APAR IS NOT APPLIED
SECURITY EXIT NOT INSTALLED	N/A	5 DEFAULT OTMA COMMANDS ALLOWED	NOT INVOKED	NOT INVOKED IF APAR NOT APPLIED, RACF MAKES FINAL DECISION IF APAR APPLIED, NO RACF CHEKING DONE	NOT INVOKED IF APAR NOT APPLIED, RACF MAKES FINAL DECISION IF APAR APPLIED, NO RACF CHECKIN G DONE	NOT INVOKED IF APAR NOT APPLIED, RACF MAKES FINAL DECISION IF APAR APPLIED NO RACF SECURITY CHECKING DONE
DFSCCMD0 INSTALLED	N/A	**INVOKED** ADDITIONAL (non-default) COMMANDS ALLOWED	N/A	N/A	N/A	N/A
DFSCTRN0 INSTALLED	N/A	N/A	**INVOKED** EXIT MAKES FINAL DECISION	**INVOKED** May be overridden by DFSCTSE0	**INVOKED** May be overridden by DFSCTSE0	**INVOKED** May be overridden by DFSCTSE0
DFSCTSE0 INSTALLED	N/A	N/A	**NOT INVOKED** FOR SOURCE TRANSACTION	**INVOKED** EXIT MAKES FINAL DECISION	**INVOKED** EXIT MAKES FINAL DECISION	**INVOKED** EXIT MAKES FINAL DECISION

OTMASE=C or /SECURE OTMA CHECK

When the OTMA security level is **CHECK**, RACF **is** invoked by IMS/OTMA. Like the OTMA security level of NONE, **CHECK** (or C) is also an ***IMS-wide*** security level. An IMS-wide level means that IMS takes the same action for each message received via OTMA. When the OTMA security level is **CHECK**, IMS **invokes** RACF to perform all of the following:

- Client-bid security checking for client-bid messages.
- Userid validation and ACEE creation for OTMA client applications and end user userids.
- IMS command authorization for command messages received via OTMA.
- IMS transaction authorization for transaction input messages received via OTMA.
- Authorization checking for subsequent IMS resources (such as transactions, databases, segments, fields, or other resources) requested during source transaction processing when the application issues a CHNG call, an AUTH call, and/or does a deferred conversational program-to-program message switch.

IMS/OTMA Client-bid Security Checking For Security Level CHECK

Before an OTMA client such as IMS Connect can pass end user messages to IMS using the OTMA interface, the OTMA client must first join the same XCF group as the IMS to which it intends to communicate. Next the OTMA client sends a ***'client-bid'*** message to IMS/OTMA to request a connection to that IMS. The client-bid is a special type of message that must be the ***first*** message sent by the OTMA client to IMS/OTMA. The security information pertaining to the OTMA client subsystem is contained in the SECURITY DATA (SE) portion of an OTMA message prefix. Figure 3, *'Security Data (SE) Portion of a Client-Bid Message'*, shows the portion of a client-bid message that contains the security-related information.

Figure 3. SECURITY DATA (SE) PORTION OF A CLIENT-BID MESSAGE

FLOW	SECTION	CONTENT OF PREFIX SECTION
CLIENT-BID	MC	MESSAGE TYPE=COMMAND, COMMAND TYPE=CLIENT-BID, ...
	SD	MEMBER NAME=HWSMEM, ACEE AGING VALUE, HASH TABLE SIZE, ...
	SE	SECURITY FLAG (N | C | F) UTOKEN USERID SAF PROFILE

UTOKEN IS REQUIRED FOR CLIENT-BID

USERID AND SAF PROFILE NOT USED WHEN OTMA MESSAGE HAS UTOKEN

When an OTMA security level of **CHECK** is used, the client-bid security check is performed as a result of IMS/OTMA receiving the client-bid message. **When an OTMA security level of CHECK is used,** OTMA clients can not connect to IMS until after RACF has been invoked to perform authorization checking for the client-bid request. Upon receipt of the client-bid message, IMS/OTMA retrieves the UTOKEN from a field in the SECURITY DATA (SE) section of one of the OTMA message prefixes. When an OTMA security level of **CHECK** is used, the UTOKEN is ***REQUIRED*** in each client-bid message received by IMS/OTMA. The presence of a UTOKEN in the client-bid message indicates to IMS that the userid in the UTOKEN in the bid message has been previously verified by an authorized subsystem, such as z/OS or OS/390. **If the incoming client-bid message does not contain a UTOKEN, the client-bid is rejected and the OTMA client can not communicate with IMS/OTMA.**

Upon receipt of a client-bid message with a UTOKEN, IMS/OTMA invokes RACF twice to determine if the OTMA client is authorized to connect to IMS/OTMA.

- First, IMS/OTMA invokes RACF to build an Accessor Environment Element (ACEE), a RACF security control block, from the UTOKEN in the client-bid message. The ACEE is required by RACF in order to perform the connection request.

- After RACF returns the ACEE, IMS/OTMA invokes RACF a second time to determine if the OTMA client is allowed to connect to (and communicate with) IMS/OTMA.

An example of the first RACROUTE macro issued by IMS/OTMA to invoke RACF is illustrated below.

Example of an IMS/OTMA SAF call for OTMA client-bid verification. Before RACF can perform the connection (client-bid) security check, a RACF security control block called an ACEE must be built. The following RACROUTE is an example of the call that IMS/OTMA issued to invoke RACF to build the ACEE.

RACROUTE REQUEST=VERIFY,ENVIR=CREATE,WORKA=(2),TOKNIN=(3), PASSCHK=NO,...

The address of the UTOKEN in the client-bid message is provided by the register identified on the **TOKNIN=** keyword. In the example register 3 (R3) provides the address of the UTOKEN.

After RACF has built an ACEE and returned the address of the ACEE for the OTMA client, the next RACF check can be made. The second RACROUTE macro is issued by IMS/OTMA to invoke RACF for client-bid (connection) security. The RACROUTE macro contains the name of the following:

- Address of the ACEE (**ACEE=)** to be used in RACF authorization processing.
- RACF resource class (**CLASS=)** to check for a security profile.
- Name of the security profile (**ENTITY=)** to locate within the resource class that is used in client-bid authorization processing .

RACF searches the resource class for a profile by that name or for a profile that satisfies the profile naming conventions. Refer to the example below to see how IMS/OTMA uses the System Authorization Facility (SAF) interface to invoke RACF when issuing the RACROUTE macro for client-bid authorization processing.

Example of an IMS/OTMA SAF call for client-bid security checking:

RACROUTE REQUEST=AUTH,**ACEE=acee_address**, WORKA=(2),**CLASS=FACILITY**, **ENTITY=IMSXCF.XCFGROUP.XCFMBNM,...**

Note that the CLASS= keyword specification is used to supply the name of the RACF resource class (in this case, the FACILITY class).

Likewise, the ENTITY= keyword specification is used to supply the name of the profile (within the FACILITY class) to use for authorization checking. In the example, the name of the profile is IMSXCF.XCFGROUP.XCFMBNM.

FACILITY Class Profile Naming Conventions

When RACF administrators create security profiles to secure the client-bid connection requests to IMS/OTMA, they need to be aware of the profile naming convention used by IMS. The naming convention used by IMS is as follows:

- The high level qualifier in the FACILITY class profile name **must** be '**IMSXCF**'.

- The second level qualifier in the profile name is the name of the **XCF group** that both IMS and the OTMA client joined during subsystem initialization. In the example above, the second level qualifier and the name of the XCF group that IMS and the OTMA client joined is '**XCFGROUP**'. You can identify the name of the XCF group that IMS joined by either of the following methods:

  1. Issue the **/DISPLAY OTMA** command. The XCF group name will be displayed in the output response.

  2. Examine the value specified on the **GRNAME=** keyword in IMS startup parameters. IMS startup parameters are located in the DFSPBxxx member in the procedure library [IMS.PROCLIB(**DFSPBxxx**)].

- The low level qualifier in the profile name is the **XCF member name of the *OTMA client***. In the above example, the XCF member name and low level qualifier is '**XCFMBNM**'.

  Like IMS, the OTMA client specifies its XCF member name as a startup/execution parameter. In the case of IMS Connect, the XCF member name for IMS Connect is specified by the value on the **MEMBER=** keyword on the DATASTORE statement in the HWSCFG file containing the IMS Connect startup parameters.

RACF administrators should create FACILITY class profile names that conform to the naming convention.

Authorizing OTMA Client-bid Requests

If you want to limit the OTMA clients that can connect to IMS/OTMA to a specific set of OTMA clients, create one or more security profiles in the RACF FACILITY class and authorize those specific OTMA clients. The RACF command samples shown below could be used to secure the client-bid connection requests to IMS/OTMA. Note that the **USERID** (not the XCF member name) of the OTMA client is authorized (using the PERMIT command) with **ACCESS(READ)** or higher.

```
RDEFINE IMSXCF.XCFGROUP.HWSMEM UACC(NONE)
PERMIT IMSXCF.XCFGROUP.HWSMEM CLASS(FACILITY) ID(HWS1PROD) ACCESS(READ)
```

NOTES:
1. The RDEFINE command is used to create a security definition or profile. In the example, the profile name is IMSXCF.XCFGROUP.HWSMEM. The profile has a universal access level (UACC) of NONE.

2. The PERMIT command is used to authorize one or more OTMA clients to access IMS/OTMA. The above RACF definition assumes that the userid for OTMA client, in this case, HWS1PROD, has already been defined to RACF.

When RACF is invoked to check the FACILITY class profile, RACF determines whether or not the profile exists in the FACILITY class. If a discrete profile (or a generic profile that satisfies the profile naming convention) is found in the FACILITY class, RACF determines whether or not the OTMA client's userid (or group) is in the list of authorized userids associated with the profile. Remember, the PERMIT command supplies the list of groups/userids by the ID(...) parameter and the access level by the ACCESS(...) parameter.

In response to the IMS/OTMA RACROUTE request, RACF sends a return code to IMS/OTMA which indicates the result of the authorization check. The return code is typically one of the following:

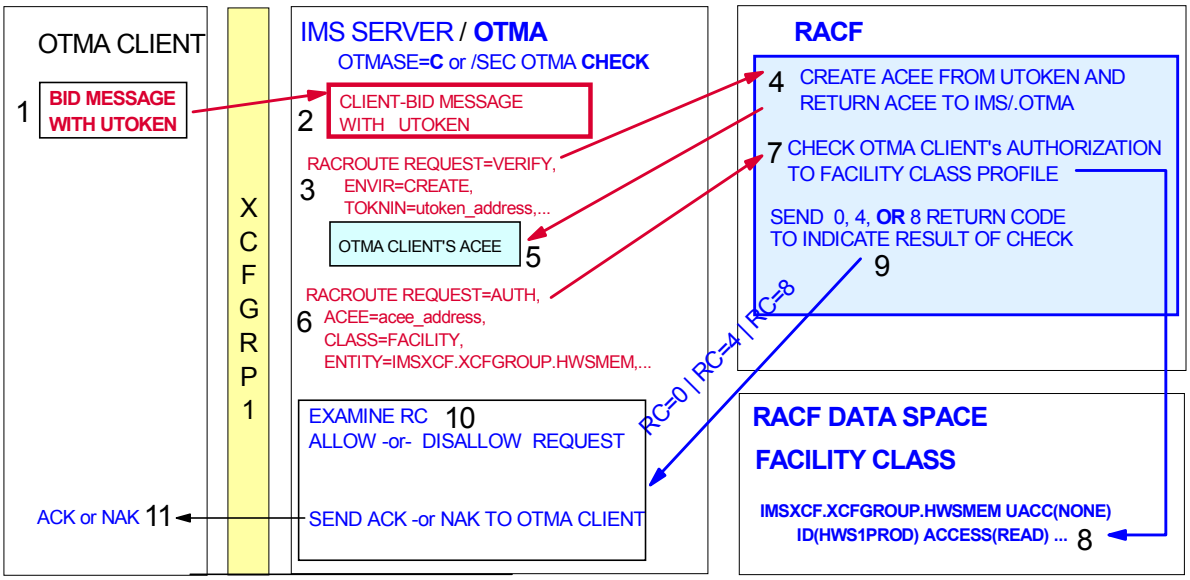
- Return code **0** which indicates that a profile was found and the OTMA client's userid (or group) is authorized. In this case IMS/OTMA accepts the client-bid request and sends an ACK (for acknowledgment) message to the OTMA client to indicate a successful client-bid.

- Return code **4** which indicates that a profile was **not** found. By default, IMS assumes that the installation did not want to secure the connection request since a profile was not defined to secure the connection. Therefore IMS accepts the client-bid request and sends an ACK to the OTMA client.

- Return code **8** which indicates that a profile was found, but the OTMA client's userid (or group) is not authorized to connect to IMS/OTMA. In this case IMS/OTMA rejects the client-bid request and send a NAK (for negative acknowledgment) message to the OTMA client.

Figure 4 OTMA Client-Bid Security Checking summarizes the connection request security checking described above.

Figure 4. OTMA Client-Bid Security Checking



1. The OTMA client sends a client bid message to IMS/OTMA. When the OTMA security level is **CHECK**, the bid message must contain a UTOKEN.
2. If the client-bid message does not contain a UTOKEN the client-bid is rejected when the OTMA security level is **CHECK**.
3. RACF is invoked to create an ACEE for the OTMA client. The address of the UTOKEN to use to create the ACEE is provided by the **TOKNIN** keyword.
4. RACF creates an ACEE from the UTOKEN and returns the ACEE to IMS/OTMA.
5. IMS/OTMA saves the ACEE for the OTMA client in the IMS control region.
6. IMS/OTMA invokes RACF to determine if the OTMA client is authorized to connect to IMS/OTMA. The ACEE of the OTMA client is supplied by the **ACEE=** keyword.
7. RACF performs authorization checking using the the in-storage profiles in the FACILITY class (8.) to determine if the userid (or RACF group) of the OTMA client is authorized to connect to IMS/OTMA.
9. RACF sends a return code to IMS/OTMA to indicate the result of the authorization check. Either a 0, 4, or 8 return code is sent to IMS/OTMA.
10. IMS/OTMA examines the RACF return code. The connection request (client-bid) is granted when the RACF return code is 0 or 4, whereas the connection request is denied when the RACF return code is 8.
11. IMS/OTMA sends the OTMA client an ACK if the client-bid connection request is granted. A NAK is sent to the OTMA client if the client-bid connection request is denied.

As previously mentioned, the presence of a UTOKEN in the incoming client-bid message indicates to IMS that the userid in the message has been previously verified by an authorized subsystem, such as z/OS or OS/390. If the incoming client-bid message does not contain a UTOKEN, the client-bid is rejected when the OTMA security level is CHECK.

If the security check indicates the OTMA client is authorized (RC=0 or RC=4), IMS allows the OTMA client's client-bid request. **Then IMS/OTMA builds a 'hash table' for the authorized OTMA client.** Finally, IMS/OTMA sends an ACK message to the OTMA client to indicate that the client-bid request had succeeded. Upon receipt of the ACK message, the OTMA client can begin sending TCP/IP end user messages to IMS/OTMA for processing.

If the security check indicates the OTMA client is **not** authorized, IMS/OTMA does all of the following:

- Rejects the client-bid request.
- Does **not** build a hash table for the OTMA client.
- Sends a NAK message to the OTMA client to indicate an **unsuccessful** client-bid.

Userid Validation and OTMA Client Hash Tables

The hash tables, located in IMS control region storage, are used to minimize the number of calls to RACF to validate/verify the userids in subsequent TCP/IP **end user messages** that will be transmitted to IMS/OTMA by OTMA clients. OTMA maintains a

hash table for each OTMA client. Each OTMA client's hash table will eventually be populated with userid entries that have been previously verified by RACF. For example, an OTMA client like IMS Connect may have previously verified the userid.

IMS/OTMA invokes RACF to verify the userid in each message received from an end user upon receipt of the message when the OTMA security level is CHECK. RACF is used to verify the userid only, the end user's password is not verified because it is not passed to OTMA in incoming message from the OTMA client. As with client-bid messages, if the end user message contains both a UTOKEN field and a userid field, the UTOKEN is used in the RACF verification request rather than the userid because the UTOKEN also contains the userid. If the userid has been defined to RACF an ACEE is returned as part of the verification request.

Once an end user's userid has been validated by IMS/OTMA, the userid and a pointer to the ACEE returned by RACF is stored in the OTMA client's hash table. Using the hash table, IMS/OTMA does not have to call RACF to validate the same userid each time an incoming message is received from the same userid. For example, a particular userid could send hundreds (or thousands) of input messages to IMS/OTMA in a short time span. It would be inefficient for IMS/OTMA to call RACF hundreds (or thousands) of times to validate the same userid. Each OTMA client's hash table is used to optimize IMS/OTMA-RACF validate/verify processing.

It should be noted that each OTMA client indicates to IMS, through use of an ***ACEE aging value*** on the client-bid message, how long an end user's ACEE is considered valid in IMS. For example, IMS Connect specifies a fixed value of about two billion seconds as the ACEE aging value on the client-bid request message. This indicates that ACEEs built for end user userids accessing IMS/OTMA through IMS Connect may remain in the hash table for up to two billion seconds.

On the other hand, MQSeries allows the installation to specify the ACEE aging value on the client-bid when the MQ-IMS Bridge application is used for transmitting message to IMS/OTMA. The installation may specify a value between one second and about two billion seconds. Therefore, the duration of end user userid entries in the hash table for the OTMA MQSeries-IMS Bridge application is between one second and two billion seconds, as specified by MQSeries on the client-bid request message.

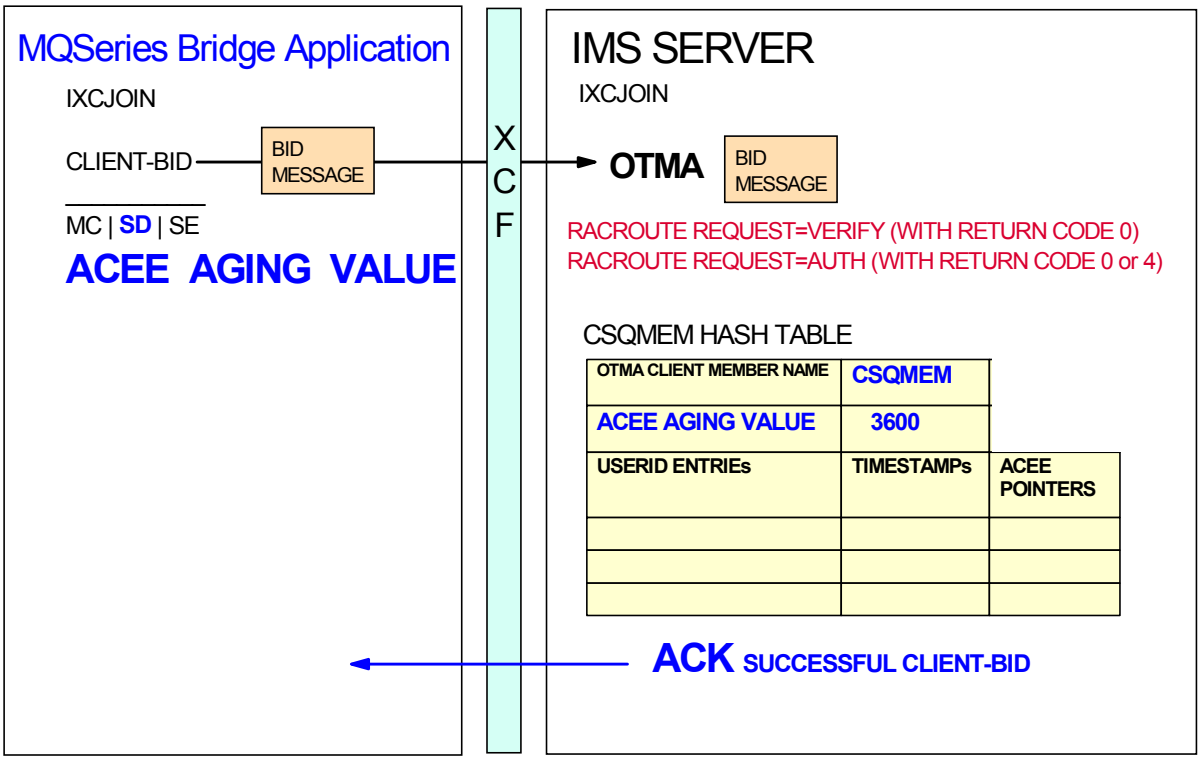
Whether a message was received via IMS Connect, MQSeries, or other OTMA clients, each time IMS/OTMA receives a message, a check is made for an end user's userid entry in the OTMA client's hash table prior to invoking RACF to verify the userid. If an end user's userid entry is found, IMS/OTMA calculates the length of time the entry has been in the hash table. The calculation can be made by subtracting the timestamp in the hash table entry from the current time and comparing the result with the ACEE aging value specified by the OTMA client on the client-bid. If the final result exceeds the ACEE aging value for the OTMA client, IMS/OTMA must invoke RACF to reverify the userid and return a new ACEE (as the older ACEE is no longer valid). If the final result does not exceed the ACEE aging value for the OTMA client, IMS can avoid the call to RACF to verify the userid and use the end user's existing ACEE to invoke RACF for command or transaction authorization processing. IMS follows the pointer in the hash table entry to the ACEE and supplies the ACEE address on the subsequent RACROUTE REQUEST=FASTAUTH macro issued to invoke RACF to perform command or transaction authorization.

If the userid entry has been in the hash table longer than the ACEE aging value specified by the OTMA client on the client-bid, IMS/OTMA does not follow the pointer to the ACEE. Rather, IMS/OTMA must invoke RACF to reverify the userid. And, if verified by RACF, IMS/OTMA must invoke RACF again to perform command or transaction authorization processing.

Refer to *Figure 3, 'SECURITY DATA (SE) PORTION OF A CLIENT-BID MESSAGE*'. The STATE DATA (SD) portion of the client-bid message in the OTMA prefix contains the ACEE aging value specified by the OTMA client.

When the OTMA client has successfully connected to IMS/OTMA, IMS/OTMA creates a hash table for the OTMA client. *Figure 5, 'OTMA Hash Table Creation'* illustrates when a hash table is created for an OTMA client (e.g. MQSERIES-IMS Bridge). *Figure 6, 'Hash Table Use'* illustrates how the hash table is used.

Figure 5. Hash Table Creation



As Figure 5 illustrates, an MQSeries-IMS Bridge application (XCF member CSQMEM) joins the same XCF group as IMS. The MQSeries-IMS Bridge application issues the client-bid message. A successful client-bid (which means the RACF return code was either 0 or 4) results in IMS doing both of the following:

- Creating a hash table for the OTMA client, as in this example, where a hash table is created for MQSeries-IMS Bridge OTMA client CSQMEM.
- Issuing the ACK to the OTMA client to indicate the connection has been successfully established.

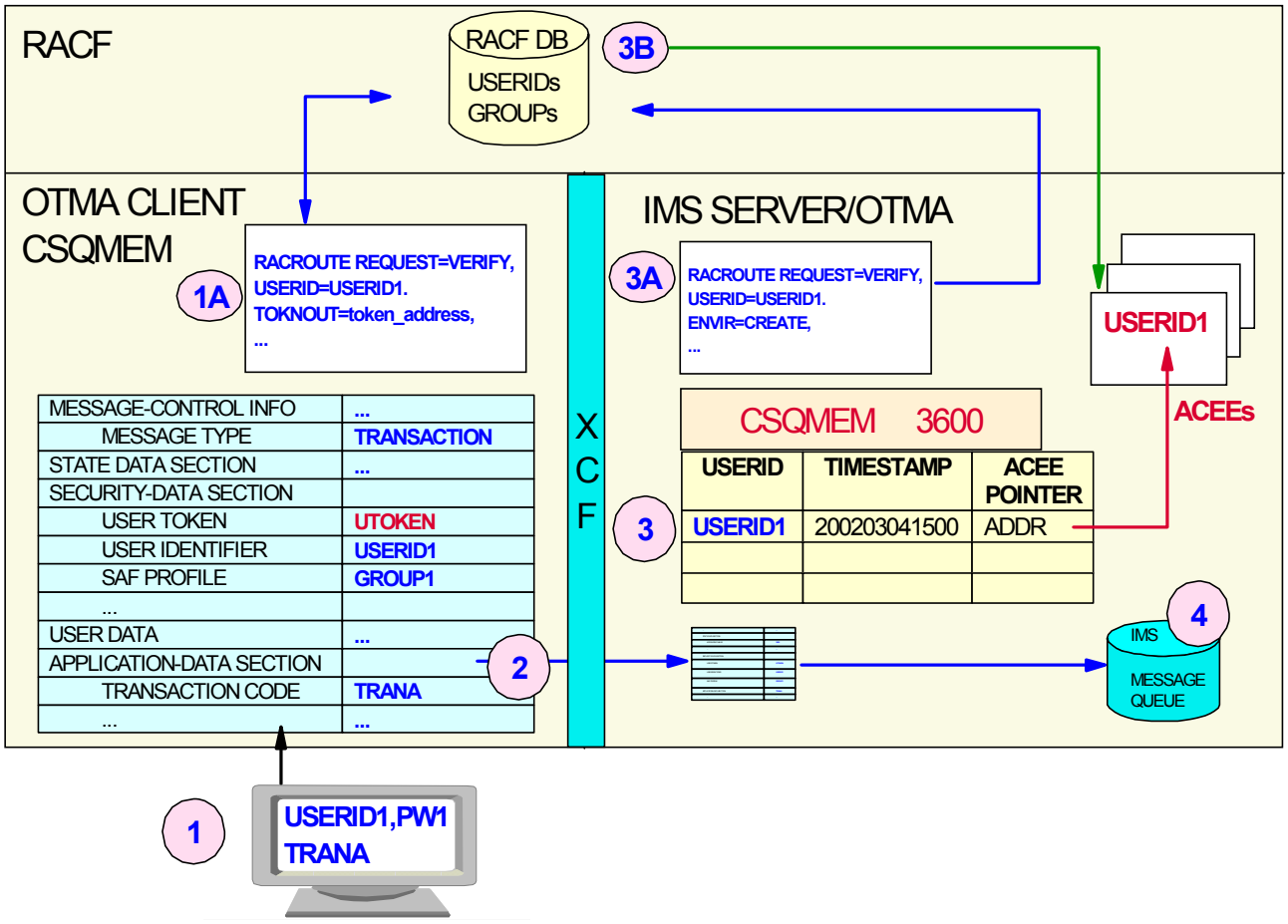
The aging value 3600 seconds in the hash table indicates that the end user userid entries which will be placed in CSQMEM's hash table are valid for 3600 seconds (1 hour). After a successful client-bid, the hash table is empty and does not contain userid entries. But once the client-bid has succeeded, MQSeries may begin transmitting end user messages to IMS/OTMA.

IMS transaction and IMS command messages submitted by end users contain the same (or similar) security information as the security information in the client-bid message. As end user messages are transmitted to IMS/OTMA, RACF (or an equivalent) will be invoked to verify the end user userids in the incoming messages; or to build ACEEs for end user messages containing UTOKENs. An entry will be placed in the OTMA client's hash table for the userid after RACF has verified the userid and returned an ACEE for the userid.

Through use of the OTMA client's hash table, upon receipt of subsequent messages from the same userid, IMS is able to avoid calls to RACF to verify the userids in incoming messages if the aging value has not been reached.

Figure 6, *'Hash Table Use'* illustrates how CSQMEM's hash table is used to minimize RACF userid validation processing.

Figure 6. Hash Table Use



The major action(s) taken at each step are listed below.

1. Once the connection has been established between the OTMA client (e.g. CSQMEM) and IMS/OTMA, end users may transmit messages through the OTMA client to IMS/OTMA. In the example, USERID1 enters a message requesting the execution of TRANA. The message is transmitted to the OTMA client.

   Once the OTMA client CSQMEM has received the message, the message is reformatted so that it will be acceptable input to OTMA. For example, TCPIP headers are removed, the message contents are translated from ASCII to EBCDIC, and OTMA headers (prefixes) are built.

1A. If the OTMA client has been configured to verify the userid (and optionally, the password for the userid, such as PW1) in the incoming message, the OTMA client invokes RACF. The RACROUTE REQUEST=VERIFY call made to RACF requests that RACF return a UTOKEN for a verified userid. For ***verified*** userids, RACF returns a UTOKEN to the OTMA client. The OTMA client places the UTOKEN in the message that is destined for IMS/OTMA.

If the OTMA client has ***not*** been configured to verify the userid (and optionally, the password) in the incoming message, the OTMA client places the unverified userid **(without the password)** in the message that is destined for IMS/OTMA.

2. The reformatted message with the OTMA headers/prefixes is transmitted to IMS/OTMA using XCF communications services. NOTE THAT THE PASSWORD (e.g. PW1) IS NOT SENT TO OTMA IN THE MESSAGE.

3. Upon receipt of the message, IMS checks the OTMA client's hash table to see if there is a userid entry in the table corresponding to the userid in the incoming message (in this example USERID1). **When IMS/OTMA receives messages that do not contain either a UTOKEN nor a userid, the message is rejected when the OTMA security level is CHECK.**

   If IMS/OTMA finds an entry for the userid (e.g. USERID1) in the OTMA client's hash table, IMS/OTMA calculates (using the timestamp, current time, and ACEE aging value) whether the userid has been in the hash table for longer than the ACEE aging value (3600 seconds) associated with the OTMA client's hash table. If the userid entry in the hash table has not expired, IMS follows the pointer to the ACEE for that userid (USERID1) and invokes RACF to determine if the userid (USERID1) is authorized to execute the requested resource (TRANA). **Thus, IMS/OTMA is able to eliminate the calls to RACF that are described in steps 3A and 3B.**

3A. If a userid entry for USERID1 is not found in the OTMA client's hash table, IMS must invoke RACF to verify the userid in the incoming message and to return an ACEE for a verified userid. Recall that before IMS can invoke RACF to check USERID1's authority to execute TRANA, and ACEE must exist. The RACROUTE REQUEST=VERIFY call to RACF causes RACF to return an ACEE for USERID1, provided that USERID1 has been defined in the RACF database. The process of returning an ACEE for a verified userid is very efficient if the OTMA client pre-verified the userid and placed a UTOKEN in the message prior to sending the message to IMS/OTMA.

3B. Without the UTOKEN, RACF may have to perform an I/O to the RACF database to retrieve the user and group profiles for USERID1.

4. **After an ACEE has been returned for a verified userid, IMS invokes RACF a second time to determine if the userid (USERID1) in the message is authorized to execute the resource requested in the message (TRANA).** This step is not shown in the Figure 6. If the userid (USERID1) was successfully verified and that same userid is authorized to the requested resource (TRANA), IMS queues the transaction on the message queue for processing (or executes the IMS command if the message contained an IMS command instead of a transaction code).

As Figure 6 illustrates, the use of an OTMA client's hash table minimizes the number of times RACF is invoked to perform userid validation/verification. The hash table for each OTMA client can have up to 5,000 entries. Each OTMA client's hash table is created in the IMS control region during client-bid processing. Each OTMA client's hash table is deleted when the connection between the OTMA client and IMS/OTMA is stopped.

Some OTMA clients, such as IMS Connect, provide a command to stop the connection between the OTMA client and IMS/OTMA so the OTMA client's hash table can be deleted and rebuilt. This is helpful when the security administrators have made a number of changes to RACF user and group profiles and they want the changes reflected in IMS resource authorization processing.

Other OTMA client's do not provide a command to stop the connection. In order to delete the OTMA client's hash table and cause it to be rebuilt, OTMA may need to be stopped and restarted. Stopping and restarting OTMA causes all hash tables for all OTMA client's to be deleted and rebuilt. There are two important points here:

1. If security administrators make lots of changes to user and group profiles and the updated profile information is required to be current in IMS/OTMA, make sure that the OTMA client used to communicate with IMS/OTMA can meet the requirements of the installation with respect to stopping and restarting the connection between the systems.
2. Make sure the ACEE aging value used can reflect changes made by security administrators if this is important to the installation.

IMS/OTMA Command Authorization Checking For Security Level CHECK

When the OTMA security level is **CHECK**, IMS/OTMA calls RACF to determine if the userid in the incoming message is authorized to execute the command in the incoming message. On the RACROUTE request call to RACF, IMS supplies a pointer to the ACEE for the userid; the RACF class name; and the security profile name to use for command authorization processing (see the diagram below).

```
RACROUTE REQUEST=FASTAUTH,USERID=USERID1,
ACEE=acee_address,CLASS=CIMS,ENTITY=DIS,
ATTR=READ,...
```

RACF checks the profiles in a RACF data space in the CIMS and DIMS classes (or the equivalent command resource classes) to see if a profile exists to secure the command. If a profile is found to secure the command, RACF determines if the userid is authorized to execute the command. In the above example, IMS invokes RACF to perform resource authorization checking (**REQUEST=FASTAUTH**) to determine the following:

- Whether USERID1 (or a RACF group that USERID1 is a member of) is authorized to execute the /DISPLAY command. The userid is supplied by the **USERID=** keyword and the address of the ACEE for that userid is supplied by the **ACEE=** keyword.
- Whether the CIMS resource class contains a discrete or generic profile to protect the /DISPLAY command.

  The resource class that RACF is to use is supplied by the **CLASS=** keyword and in the example the resource class is CIMS.

  The name of the profile in the resource class is supplied by the **ENTITY=** keyword. In the example the profile name is **DIS**, which is the 3 character profile name for the /DISPLAY command.

The security administrator issues RACF commands to create security profiles and access lists for IMS commands. For example, the following RACF commands may be issued to secure the IMS /DISPLAY command and authorize userids that are connected to GROUP1 to execute the command. If USERID1 is connected to the RACF group GROUP1, then USERID1 is allowed to execute the /DISPLAY command.

```
RDEFINE CIMS DIS OWNER(IMSADMIN) UACC(NONE)
PERMIT DIS CLASS(CIMS) ID(GROUP1) ACCESS(READ)
```

The result of the RACF command authorization check is returned to IMS in the form of a return code; either 0, 4, or 8, where:

- **0** indicates the userid is authorized to execute the command.
- **4** indicates that no profile exists to secure the command, in which case IMS allows the command to execute.
- **8** indicates the userid is not authorized to execute the command.

If your installation has also included the Command Authorization Exit Routine (DFSCCMD0) in your IMS system, IMS will invoke the exit after receiving the RACF return code. The RACF return code is passed to DFSCCMD0 in the parameter list. DFSCCMD0 may be coded to agree with the RACF decision in some cases, while overriding the RACF decision in other cases; the logic in the exit is strictly up to your installation. In any event, DFSCCMD0 makes the final decision on whether or not the userid will be allowed to execute the command.

IMS/OTMA Transaction Authorization Checking For Security Level CHECK

When the OTMA security level is **CHECK**, IMS/OTMA calls RACF to determine if the userid in the incoming message is authorized to execute the transaction in the incoming message. IMS supplies a pointer to the ACEE for the userid; the RACF class name (TIMS or equivalent); and the profile name (TRANA) for the transaction on the RACROUTE REQUEST=FASTAUTH call to RACF (see the diagram below).

```
RACROUTE REQUEST=FASTAUTH,USERID=USERID1,
ACCE=acee_address,CLASS=TIMS,ENTITY=TRANA,ATTR=READ,...
```

RACF checks the in-storage profiles in the TIMS and GIMS (or equivalent classes) to see if a security profile exists to secure the transaction TRANA. If a profile is found to secure the transaction, RACF determines if the userid is authorized to execute the transaction.

The security administrator issues RACF commands to create security profiles and access lists for IMS transactions. For example, the following RACF commands may be issued to secure the IMS transaction TRANA and authorize userids that are connected to GROUP1 to execute the transaction. If USERID1 is connected to the RACF group GROUP1, then USERID1 is allowed to execute TRANA..

```
RDEFINE TIMS TRANA OWNER(IMSADMIN) UACC(NONE)
PERMIT TRANA CLASS(TIMS) ID(GROUP1) ACCESS(READ)
```

The result of the RACF transaction authorization check is returned to IMS in the form of a return code; either 0, 4, or 8, where:

- 0 indicates the userid is authorized to execute the transaction
- 4 indicates that no profile exists to secure the transaction, in which case IMS allows the transaction to execute
- 8 indicates the userid is not authorized to execute the transaction.

Transaction Authorization Exit Routine (DFSCTRN0)

If your installation has also included the Transaction Authorization Exit Routine (DFSCTRN0) in your IMS system, IMS ***may*** or ***may not*** invoke the exit after receiving the RACF return code.

If IMS receives a return code of 0 or 4 from RACF, IMS invokes DFSCTRN0; resulting in the exit having the final decision on whether or not the userid can execute the *incoming* transaction. The RACF return code is passed to DFSCTRN0 in the parameter list. DFSCTRN0 may be coded to agree with the RACF decision in some cases, while overriding the RACF decision in other cases. The logic in the exit is strictly up to your installation. In any event, DFSCTRN0 makes the final decision on whether or not the userid will be allowed to execute the transaction.

However, if IMS receives a return code of 8 from RACF; IMS does ***not*** invoke the Transaction Authorization Exit (DFSCTRN0). The incoming transaction message is rejected and the transaction is ***not*** queued to the message queue for processing. When a return code 8 is returned, the DFS1292 security violation message is sent to the OTMA client.

Dependent Region Security Checking

If the security checks done by RACF, and optionally, the Transaction Authorization Exit Routine (DFSCTRN0) result in successful authorization of the userid in the input message, IMS queues the message to the message queue. From this point on, for the OTMA security level of **CHECK**, the security checking that takes place is business as usual. It does not matter that the transaction entered IMS via OTMA.

The application program issues a DL/I Get Unique (GU) call to retrieve the message from the message queue for processing.

Sometimes the application program in the dependent region will take an action while processing the message that will result in a transaction authorization check. For example, while processing the message, if the application does one or more of the following, IMS may require a transaction authorization security check:

- Issue a CHNG call with the destination set as a transaction code.
- Issue an AUTH call with the CLASSNAME specified as TRAN.
- Perform a deferred conversational program-to-program message switch (ISRT a SPA for a conversational transaction).

The transaction authorization check that is required as a result of the application taking one of the above actions is referred to as 'dependent region' security checking. The same security checking facilities (RACF and/or the Transaction Authorization Exit Routine) are invoked for CHNG calls, AUTH calls, and deferred conversational program-to-program message switches as for the input message. Therefore, security facilities are invoked in the following order for transactions requested via CHNG calls, AUTH calls, and deferred conversational program-to-program message switches:

- **(1) RACF** is invoked to build an ACEE (VERIFY) for the userid in IOPCB.

  The userid in the IOPCB is the same userid that the OTMA client placed in the userid field in the SECURITY DATA section of the OTMA message prefix. It should also be noted that the ACEE in the control region for the same userid is not used.

  After the ACEE is returned to IMS, **(2) RACF** is invoked to perform transaction authorization (FASTAUTH).

- If RACF authorized (return code 0 or 4) the userid to execute the transaction requested via the CHNG call, AUTH call or deferred conversational program-to-program message switch , then IMS invokes the **(3) Transaction Authorization Exit Routine (DFSCTRN0)** if the exit was included in the IMS system. DFSCTRN0 is passed the RACF return code and, as previously mentioned, the exit may override RACF's decision and deny access to the transaction.

  If RACF denies access (return code 8), IMS does ***not*** invoke the Transaction Authorization Exit Routine (DFSCTRN0).

- If your installation has included the **(4) Security Reverification Exit Routine (DFSCTSE0)** as part of the IMS system, IMS invokes this exit for transaction authorization related to all CHNG and AUTH calls, regardless of the RACF return code. DFSCTSE0 is passed the RACF or Transaction Authorization Exit Routine return code, which ever was the last to perform authorization processing for the transaction. DFSCTSE0 makes the final decision on granting or denying access to the transaction.

Security Reverification Exit Routine (DFSCTSE0)

The Security Reverification exit routine (DFSCTSE0) allows you to reevaluate transaction authorization checking on the DL/I CHNG calls and AUTH calls. This exit routine can be written as an entry point in the Transaction Authorization Exit Routine (DFSCTRN0). This means that you can use the Security Reverification Exit (DFSCTSE0) only if you also use the Transaction Authorization Exit (DFSCTRN0).

For CHNG and AUTH calls, even though RACF may have denied access (resulting in DFSCTRN0 ***not*** being invoked), the Security Reverification Exit Routine (DFSCTSE0) will be invoked for transactions requested via CHNG and AUTH calls. DFSCTSE0 is passed the return code of either RACF or DFSCTRN0, which ever executed just prior to DFSCTSE0. Thus, for CHNG and AUTH calls, DFSCTSE0 makes the final decision on whether access to the transaction will be granted or denied.

Considerations for Using OTMA Security Level CHECK

WHEN THE OTMA SECURITY LEVEL IS CHECK, RACF IS INVOKED THREE (3) TIMES FOR EACH CHNG AND AUTH CALL THAT REQUESTS AN IMS TRANSACTION CODE!!

- One time to create the ACEE (RACROUTE REQUEST=VERIFY,ENVIR=**CREATE**,...).
- One time to perform transaction authorization (RACROUTE REQUEST=**FASTAUTH**,...).
- One time to delete the ACEE (RACROUTE REQUEST=VERIFY,ENVIR=**DELETE,...**).

To illustrate the point, if the application that processes an OTMA transaction issues 30 CHNG calls that request IMS transaction codes, 92 calls would be made to RACF. Two (2) calls (VERIFY and FASTAUTH) would be made for the input transaction.

Ninety (90) additional calls would be made for the 30 CHNG calls (a VERIFY, FASTAUTH, and DELETE call to RACF for each CHNG call requesting a transaction code). This could adversely impact performance, especially when you receive a large number of transactions via OTMA. The performance may become even more degraded if you have also included the Transaction Authorization Exit (DFSCTRN0) and the Security Reverification Exit (DFSCTSE0) in your IMS system because:

- Both user exits are invoked after RACF if RACF authorized the request to execute the transaction.

- At a minimum, the Security Reverification Exit (DFSCTSE0) would be invoked after RACF for transactions requested via CHNG and AUTH calls.

This happens because the IMS control region has no idea (in advance) what an application running in a different address space (the dependent region) is going to do that might necessitate a transaction authorization check. Therefore it is necessary to build a security environment that is associated with the dependent region. For messages received via OTMA, the userid in the IOPCB of the message is used in authorization processing to build the security environment (ACEE). Unfortunately, an existing ACEE in the IMS control region for that same userid can not be used. This issue is currently under investigation by IMS/OTMA development. Security enhancements are planned to address this issue, especially the number of times RACF is invoked for transaction authorization related to CHNG and AUTH calls.

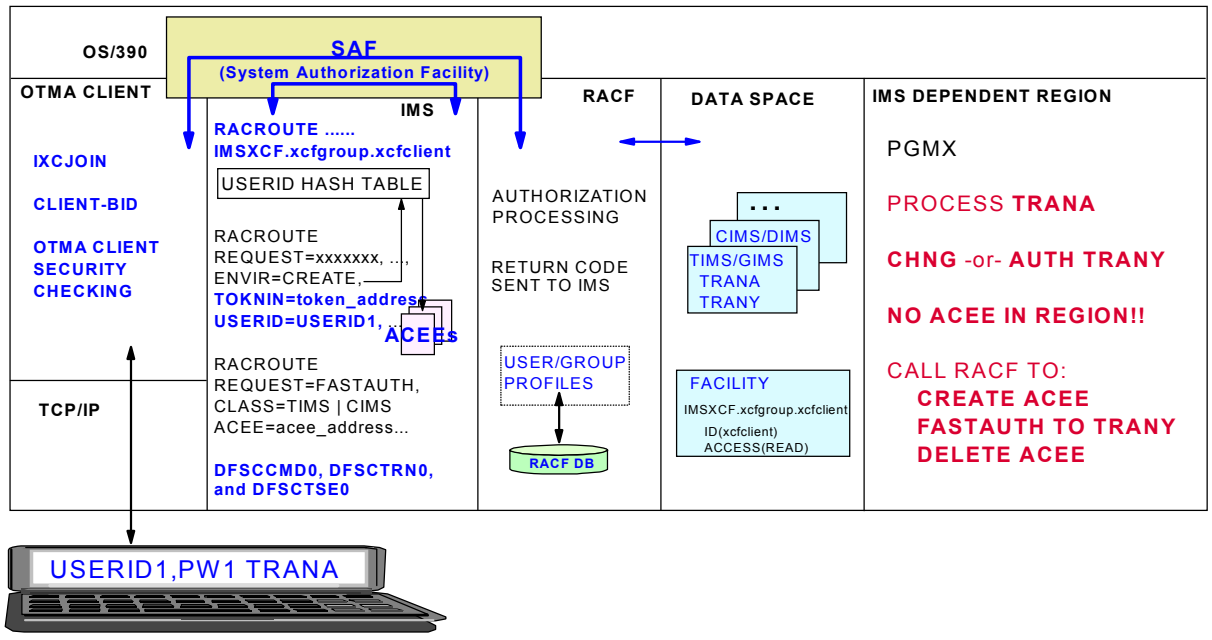
If your application issues many CHNG calls and transaction authorization is important to you; you should consider using one of the following:

- OTMA security level PROFILE.
- OTMA security level FULL.
- OTMA security level NONE.
- The Transaction Authorization Exit and optionally, the Security Reverification Exit.
- A combination of one of the above OTMA security levels in conjunction with the Transaction Authorization Exit and optionally, the Security Reverification Exit.

On the other hand, if a large number of your applications that process messages received via OTMA do not issue CHNG calls, AUTH calls, nor perform deferred conversational program-to-program message switches, a security level of CHECK may be suitable for your environment.

Figure 7, '/SECURE OTMA CHECK or OTMASE=C' summarizes the authorization checking performed when the OTMA security level is CHECK.

FIGURE 7. /SECURE OTMA CHECK or OTMASE=C



RACF IS CALLED BY IMS FOR:

1. CLIENT-BID CONNECTION SECURITY CHECKING

2. SECURITY CHECKING FOR MESSAGES (TRANSACTION and/or COMMANDS) RECEIVED VIA OTMA
   COMMANDS: RACF INVOKED TO PERFORM COMMAND AUTHORIZATION CHECKING
   CIMS and DIMS RESOURCE CLASSES (or equivalent) USED
   COMMAND ALLOWED BY IMS IF NO CIMS | DIMS PROFILE
   COMMAND AUTHORIZATION EXIT (DFSCCMD0) CALLED IF INCLUDED IN IMS SYSTEM
   TRANSACTIONS: RACF INVOKED TO PERFORM TRANSACTION AUTHORIZATION CHECKING
   TIMS and GIMS RESOURCE CLASSES (or equivalent) USED
   TRANSACTION ALLOWED BY IMS IF NOT TIMS | GIMS PROFILE

   IF TRANSACTION AUTHORIZATION EXIT (DFSCTRN0) INCLUDED IN IMS SYSTEM
   INVOKED BY IMS WHEN RACF RETURN CODE IS EITHER 0 -or 4
   NOT INVOKED BY IMS WHEN RACF RETURN CODE IS 8

   IF SECURITY REVERIFICATION EXIT (DFSCISE0) INCLUDED IN IMS SYSTEM
   INVOKED BY IMS FOR TRANSACTIONS REQUESTED VIA CHNG OR AUTH CALLS
   REGARDLESS OF THE RACF RETURN CODE

OTMA Security Level CHECK Summary

The following table summarizes the security actions taken by IMS/OTMA when the security level is **CHECK**.

OTMA SECURITY LEVEL **CHECK**	MESSAGE ORIGIN			TRAN CODE REQUESTED VIA CHNG CALL	DATABASE, SEGMENT, FIELD, OTHER, OR TRAN CODE REQUESTED VIA AUTH CALL	CONVERSATIONAL TRAN CODE REQUESTED VIA ISRT CALL (DEFERRED CONVERSATIONAL PROGRAM SWITCH)
	OTMA CLIENT	END USER				
	CLIENT-BID	IMS COMMAND	IMS TRANSACTION			
RACF	**INVOKED** AUTHORIZED OTMA CLIENT CLIENT-BIDS ARE ALLOWED	**INVOKED** BUILD ACEE FOR USERID DO COMMAND AUTH	**INVOKED** BUILD ACEE FOR USERID DO TRAN AUTH	**INVOKED** **BUILD ACEE** DO AUTH	**INVOKED** **BUILD ACEE** DO AUTH	**INVOKED** **BUILD ACEE** DO AUTHORIZATION CHECKING
SECURITY EXIT NOT INSTALLED	N/A	RACF MAKES FINAL DECISION	RACF MAKES FINAL DECISION	RACF MAKES FINAL DECISION	RACF MAKES FINAL DECISION	RACF MAKES FINAL DECISION
DFSCCMD0 INSTALLED	N/A	**INVOKED** EXIT MAKES FINAL DECISION	N/A	N/A	N/A	N/A
DFSCTRN0 INSTALLED	N/A	N/A	**INVOKED ONLY IF RACF RC IS '0' OR '4'** IF INVOKED, EXIT MAY OVERRIDE RACF DECISION	**INVOKED ONLY IF RACF RC IS '0' OR '4'** IF INVOKED, EXIT MAY OVERRIDE RACF DECISION	**INVOKED ONLY IF RACF RC IS '0' OR '4'** IF INVOKED, EXIT MAY OVERRIDE RACF DECISION	**INVOKED ONLY IF RACF RC IS '0' OR '4'** IF INVOKED, EXIT MAY OVERRIDE RACF DECISION
DFSCTSE0 INSTALLED	N/A	N/A	NOT INVOKED FOR SOURCE TRANSACTION	**INVOKED** EXIT MAKES FINAL DECISION	**INVOKED** EXIT MAKES FINAL DECISION	**INVOKED** EXIT MAKES FINAL DECISION

OTMASE=F or /SECURE OTMA FULL

When the OTMA security level is **FULL**, the security checking is very similar to the OTMA security level CHECK, therefore duplicate information will not be repeated in most cases. **FULL** (or F) is an ***IMS-wide*** OTMA security level. **FULL** is also the ***default*** value. When the OTMA security level is FULL, RACF is called to perform all of the following:

- Client-bid security checking.
- IMS command authorization for command messages received via OTMA.
- IMS transaction authorization for transaction input messages received via OTMA.
- IMS transaction authorization for subsequent transactions requested during transaction processing when the application issues a CHNG call, AUTH call, and/or does a deferred conversational program-to-program message switch.

IMS/OTMA Client-bid Security Checking For Security Level FULL

The client-bid security checking performed when the OTMA security level is FULL is identical to that performed when the OTMA security level is CHECK.

IMS/OTMA Command Authorization Checking For Security Level FULL

The command authorization security checking performed when the OTMA security level is FULL is identical to that performed when the OTMA security level is CHECK. Since all IMS commands are executed in the IMS control region, a second ACEE is not needed for dependent region security checking.

IMS/OTMA Transaction Authorization Checking For Security Level FULL

The transaction authorization security checking performed when the OTMA security level is FULL is '***similar'*** to that performed when the OTMA security level is CHECK. RACF is invoked in the same manner as with CHECK and the same rules apply. That is, the transaction exit routines (DFSCTRN0 and DFSCTSE0) are invoked as required. For example, the Transaction Authorization Exit is not invoked when RACF denies authorization to a transaction, but the Security Reverification Exit is invoked for transactions requested via CHNG and AUTH calls, regardless of the RACF return code.

The following differences apply to the OTMA security level FULL (as compared to CHECK):

1. Upon receipt of transaction messages from the OTMA clients, IMS/OTMA issues two (2) RACROUTE REQUEST=VERIFY calls to RACF for each message received when the OTMA security level is FULL. (Only one RACROUTE REQUEST=VERIFY call is issued upon receipt of the initial message when the OTMA security level is CHECK.)

   a.) The first RACROUTE REQUEST=VERIFY call is to create an ACEE (for a validated userid) in the IMS **control region**. As previously described, the ACEE returned is pointed to by the OTMA client's hash table which allows IMS to minimize the number of subsequent VERIFY request to RACF to verify the same userid that submits multiple messages.

   b.) The second RACROUTE REQUEST=VERIFY call is to create an ACEE for the same userid in the **dependent region**. The call is made immediately after the first ACEE is returned. The process of creating a second ACEE in the dependent region is very fast since all the information needed to create a second ACEE is already in storage.

2. The other difference is in the area of security checking related to transactions requested via the CHNG call, AUTH call, and/or a deferred conversational program-to-program message switch.

   As you recall with CHECK, three (3) calls had to be made to RACF for each CHNG call, AUTH call, or deferred conversational program-to-program switch. The reason three calls had to be made was because the ACEE for the userid (in transaction message) was created in the control region (not the dependent region). Since the ACEE was not in the dependent region, IMS had to do all of the following:

   - Obtain the userid from the IOPCB in the message.

- **Invoke RACF** (#1) to create an ACEE for the userid in the message (otherwise the security check would have been made using the userid associated with the dependent region.
- **Invoke RACF** (#2) to determine if the userid in the message was authorized to execute the transaction requested on the CHNG call, AUTH call, or deferred conversational program-to-program switch.
- **Invoke RACF** (#3) to delete the ACEE after the authorization check was completed.

With security level **FULL** (since the ACEE is built in the dependent region prior to processing the initial transaction) if the application issues a CHNG call, AUTH call, or does a deferred conversational program-to-program message switch, the correct ACEE is already in the dependent region. The ACEE representing the userid in the initial input message is used in transaction authorization processing. Therefore, only a RACROUTE REQUEST=FASTAUTH call needs to be made to RACF. The ACEE representing the userid in the initial input message is kept in the dependent region until the application is finished with all processing done on behalf of the initial input message. Normally, this ACEE is not deleted until the application issues a Get Unique (GU) call for the next message it will process.

Using the previous example in the OTMA security level CHECK section of this paper, if an application program issued 30 CHNG calls, with FULL, the ACEE created in the dependent region (before the initial transaction was processed) would be used for all 30 CHNG calls. By using FULL and building the ACEE in the dependent region up front, fifty eight (58) of the RACF calls made using CHECK **would be eliminated**. The table below illustrates this point.

MAXIMUM NUMBER OF TIMES RACF INVOKED FOR OTMA SECURITY LEVEL	**FULL**	**CHECK**
RACROUTE **REQUEST=VERIFY,ENVIR=CREATE**,... (UPON RECEIPT OF INITIAL INPUT MESSAGE)	1	1
RACROUTE **REQUEST=VERIFY,ENVIR=CREATE**,... (BUILD ACEE IN DEPENDENT REGION FOR VERIFIED USERID)	1	0 (NA)
RACROUTE **REQUEST=FASTAUTH,ACEE=*acee_address***,CLASS=TIMS,ENTITY=TRANX... (TRANSACTION AUTHORIZATION CHECK FOR TRANSACTION REQUESTED IN INITIAL MESSAGE)	1	1
APPLICATION PROGRAM ISSUES 30 CHNG CALLS, EACH REQUESTING AN IMS TRANSACTION CODE		
RACROUTE **REQUEST=VERIFY,ENVIR=CREATE**,**USERID=*user_id***,.... [VERIFY AND BUILD SECURITY CONTROL BLOCK (ACEE) FOR USERID IN IOPCB OF INITIAL MESSAGE]	0 (NA)	30
RACROUTE REQUEST=FASTAUTH,ACEE=acee_address,CLASS=TIMS,ENTITY=TRANY,... (TRANSACTION AUTHORIZATION CHECK FOR TRANSACTION REQUESTED VIA CHNG CALL)	30	30
RACROUTE **REQUEST=VERIFY,ENVIR=DELETE**,**USERID=user_id**,**ACEE=acee_address**,... (DELETE ACEE WHEN NO LONGER REQUIRED)	1	30
TOTAL NUMBER OF TIMES RACF INVOKED	**34**	**92**

Consideration For Using OTMA Security Level FULL

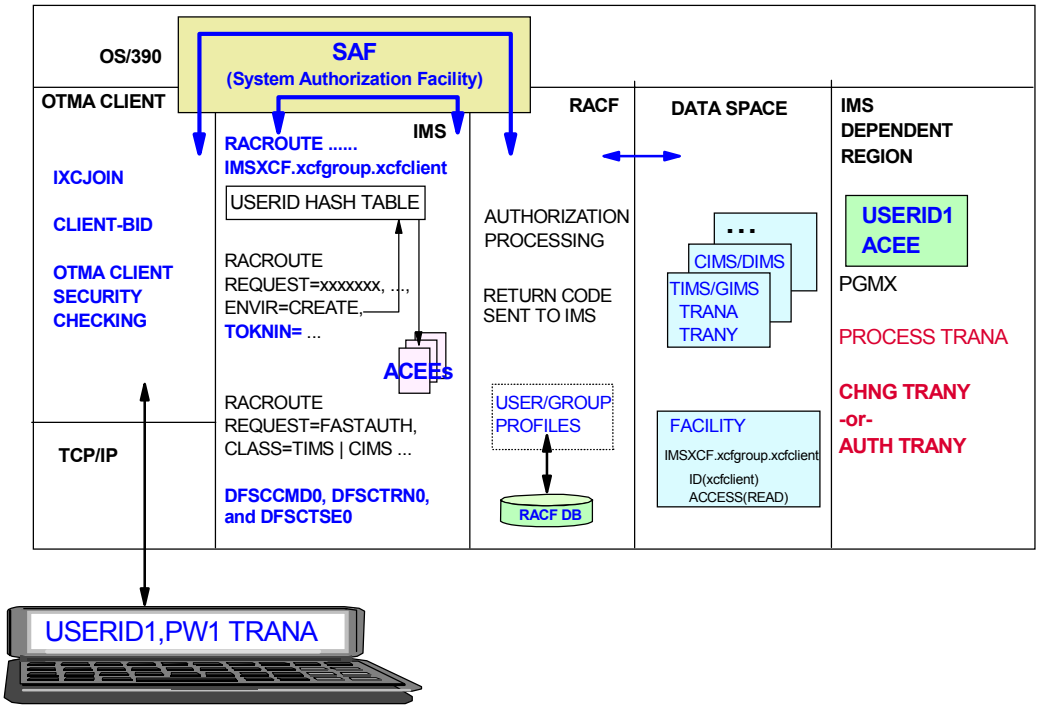
There is a consideration for using an OTMA security level of FULL. FULL will result in one additional VERIFY call to RACF to create an ACEE in the dependent region for ***every*** **transaction message** received via OTMA. Therefore, if the application does not issue any CHNG calls, AUTH calls, or perform deferred conversational program-to-program message switches; the overhead incurred (for each additional VERIFY call to create the ACEE in the dependent region) is wasted. Also, since FULL is the default, you may be unknowingly incurring the overhead if you did not specify a different OTMA security level.

If you know that a large majority of your applications that process messages received via OTMA ***do*** issue one or more CHNG calls, AUTH call, and/or perform deferred conversational program-to-program message switches, then the OTMA security level FULL may be appropriate for your environment.

On the other hand, if a large majority of your transaction requests received via OTMA are processed by applications that do not issue requests for transactions via CHNG and AUTH calls and/or ISRT calls that perform deferred conversational program switches, the OTMA security level FULL is not appropriate. In this instance, an OTMA security level of CHECK, PROFILE, or NONE is more appropriate.

Figure 8, '/SECURE OTMA FULL or OTMASE=F' summarizes the security checking performed for the OTMA security level FULL.

Figure 8. /SECURE OTMA ***FULL*** or OTMASE=***F***



RACF IS CALLED BY IMS FOR:

CLIENT-BID CONNECTION SECURITY CHECKING

SECURITY CHECKING FOR MESSAGES (TRANSACTION and/or COMMANDS) RECEIVED VIA OTMA

COMMANDS: RACF IS CALLED
CIMS and DIMS RESOURCE CLASSES USED
COMMAND ALLOWED BY IMS IF NO CIMS | DIMS PROFILE
COMMAND AUTHORIZATION EXIT (DFSCCMD0) CALLED IF INCLUDED IN IMS SYSTEM

TRANSACTIONS: RACF IS CALLED
2ND ACEE IS CREATED IN DEPENDENT REGION
TIMS and GIMS RESOURCE CLASSED USED
TRANSACTION ALLOWED BY IMS IF NO TIMS | GIMS PROFILE
TRANSACTION AUTHORIZATION EXIT (DFSCTRN0) INVOKED AFTER RACF IF RC=0 or RC=4
SECURITY REVERIFICATION EXIT (DFSCTSE0) INVOKED FOR ALL CHNG/AUTH CALLS, REGARDLESS OF RACF RC

The following table summarizes the security actions taken by IMS/OTMA when the security level is **FULL**.

OTMA SECURITY LEVEL **FULL**	MESSAGE ORIGIN			TRAN CODE REQUESTED VIA CHNG CALL	DATABASE, SEGMENT, FIELD, OTHER, OR TRAN CODE REQUESTED VIA AUTH CALL	CONVERSATIONAL TRAN CODE REQUESTED VIA ISRT CALL (DEFERRED CONVERSATIONAL PROGRAM SWITCH)
	OTMA CLIENT	END USER				
	CLIENT-BID	IMS COMMAND	IMS TRANSACTION			
RACF	**INVOKED** AUTHORIZED OTMA CLIENT CLIENT-BIDS ARE ALLOWED	**INVOKED** BUILD ACEE FOR USERID DO COMMAND AUTH	**INVOKED** **BUILD 2 ACEEs FOR USERID:** 1 IN CONTROL REGION / **2ND IN DEP. REGION** DO TRAN AUTH	**INVOKED** **USE ACEE IN DEP. REG.** DO AUTH	**INVOKED** **USE ACEE IN DEP. REG** DO AUTH	**INVOKED** **USE ACEE IN DEP. REGION** DO AUTHORIZATION CHECKING
SECURITY EXIT NOT INSTALLED	N/A	RACF MAKES FINAL DECISION	RACF MAKES FINAL DECISION	RACF MAKES FINAL DECISION	RACF MAKES FINAL DECISION	RACF MAKES FINAL DECISION
DFSCCMD0 INSTALLED	N/A	**INVOKED** EXIT MAKES FINAL DECISION	N/A	N/A	N/A	N/A
DFSCTRN0 INSTALLED	N/A	N/A	**INVOKED ONLY IF RACF RC IS '0' OR '4'** IF INVOKED, EXIT MAY OVERRIDE RACF DECISION	**INVOKED ONLY IF RACF RC IS '0' OR '4'** IF INVOKED, EXIT MAY OVERRIDE RACF DECISION	**INVOKED ONLY IF RACF RC IS '0' OR '4'** IF INVOKED, EXIT MAY OVERRIDE RACF DECISION	**INVOKED ONLY IF RACF RETURN CODE IS '0' OR '4'** IF INVOKED, EXIT MAY OVERRIDE RACF DECISION
DFSCTSE0 INSTALLED	N/A	N/A	**NOT INVOKED** FOR SOURCE TRANSACTION	**INVOKED** EXIT MAKES FINAL DECISION	**INVOKED** EXIT MAKES FINAL DECISION	**INVOKED** EXIT MAKES FINAL DECISION

OTMASE=P or /SECURE OTMA PROFILE

The other three OTMA security levels (NONE, CHECK, and FULL) are IMS-wide security levels. An IMS-wide level results in IMS taking the same security checking actions for all message received from all OTMA clients. Unlike the other OTMA security levels, **PROFILE** (or P) is ***not*** an IMS-wide security level. Rather PROFILE is a ***message-by-message*** level. Each incoming message entered through OTMA is checked to determine whether or not RACF will be called.

Each OTMA message contains a SECURITY DATA (SE) section. As you may recall, one of the fields in the SECURITY DATA section is a 1-byte security flag. The value in the security flag field determines whether or not RACF is called in IMS/OTMA when the OTMA security level is PROFILE. IMS checks each incoming message to see if the security value in incoming is set to N, C, or F.

Messages entered by end users and client-bid messages contain the 1-byte security flag field. The TCP/IP server application program (e.g. IMS Connector for Java) that builds the message for the end user can set the security flag for IMS command messages and IMS transaction messages.

It should be noted that IMS does not have to honor the security flag specification. **In fact, IMS will only honor the value in the security flag when the OTMA security level is PROFILE.** The security flag value is not checked when the OTMA security level is CHECK, FULL, or NONE. As with the other OTMA security levels, PROFILE is set as the IMS security level by issuing the command (/SECURE OTMA PROFILE) or as a startup parameter (OTMASE=P).

N, C, and F are the only valid values for the security flag field. Figure 9, '/SECURE OTMA PROFILE or OTMASE=P' shows the actions taken with respect to each of the values for the security flag.

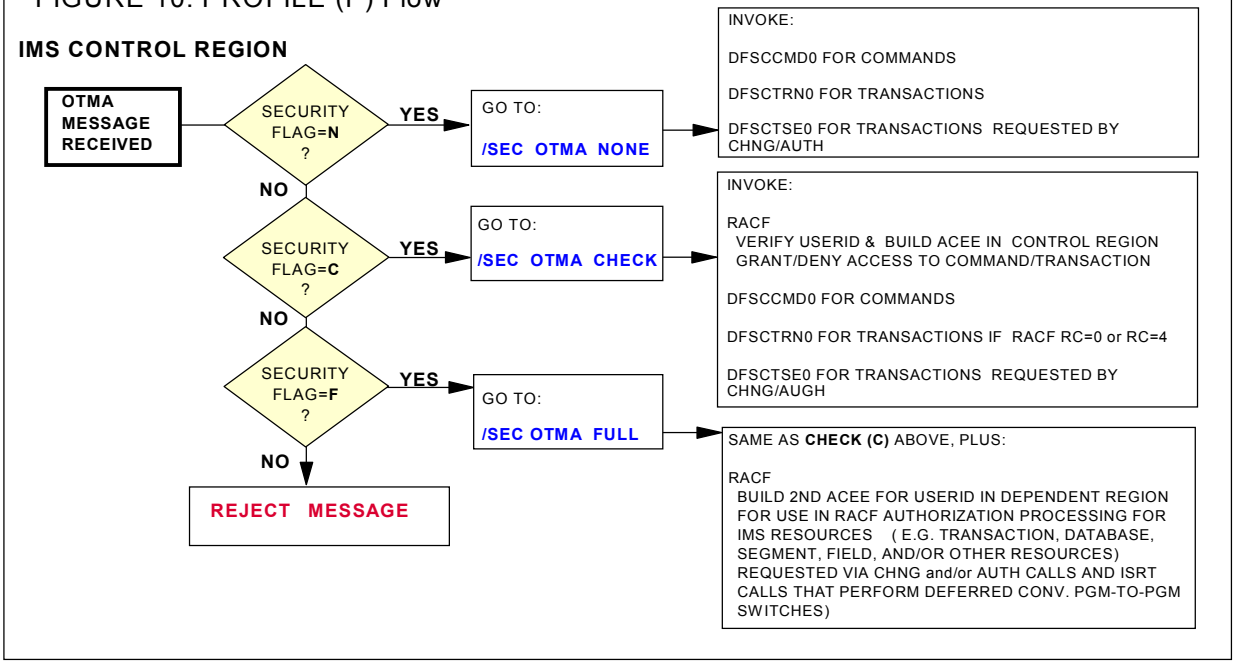
FIGURE 9. /SECURE OTMA PROFILE or OTMASE=P

BYTE	LENGTH	CONTENT	VALUE	MEANING
0	2	LENGTH		LENGTH OF SECURITY-DATA SECTION
2	1	SECURITY FLAG		
			N	NO RACF CHECKING FOR MESSAGES RECEIVED VIA OTMA
			C	RACF PERFORMS: CLIENT-BID SECURITY CHECKING USERID VERIFICATION BUILDING ACEE IN IMS CONTROL REGION TRANSACTION AND COMMAND AUTHORIZATION
			F	**RACF PERFORMS SAME AS 'C' (CHECK) PLUS:** **BUILDING 2ND ACEE IN DEPENDENT REGION TO FACILITATE FACILITATE TRANSACTION AUTHORIZATION FOR CHNG AND/OR AUTH CALL**

If you have included user security exit routines in the IMS system, they will be invoked as appropriate when the OTMA security level is PROFILE.

Refer to *Figure 10, 'Profile Flow'* for an illustration of how OTMA security works when the level has been set to PROFILE.

FIGURE 10. PROFILE (P) Flow



Considerations For Using OTMA Security Level PROFILE

One consideration for use of the OTMA security level PROFILE is that application developers set the security level as N (NONE), C (CHECK), or F (FULL) in each incoming message. You may not want application programmers deciding on whether or not RACF is called to perform security checking IMS commands and IMS transactions.

Another consideration for use of the PROFILE level is that it requires in-depth knowledge of the logic in the IMS applications that process transactions received via OTMA. You need to know, for example, whether the program issues CHNG calls and/or AUTH calls that request transactions and if so, the number of calls. You need to know these things because they have an impact on RACF-related transaction authorization calls and processing.

However, the OTMA security level of PROFILE does offer a great deal of flexibility. You may have one or more of the following situations in your installation:

- There are some IMS commands and IMS transactions, entered via OTMA, that you do not want to secure. You may want all end users to be able to execute those command and/or transactions. You could have the application programmer set the security flag to 'N' for those command and/or transactions.

- Some of the application programs that process transactions entered via OTMA clients do ***not*** issue CHNG calls, AUTH calls, nor do they perform deferred conversational program-to-program message switches. You could have the application programmer set the security flag to 'C' for IMS commands and IMS transactions where you desire RACF authorization checking for messages received via OTMA clients.

- Some of the application programs that process transactions entered via OTMA client ***do issue*** one or more CHNG calls, AUTH calls, and/or perform deferred conversational program-to-program message switches. You could have the application programmer set the security flag to 'F' for IMS transactions where you desire RACF authorization checking for transaction messages received via OTMA clients and you want to authorize the end user's userid to the destination/requested transaction on a CHNG/AUTH call issued from the dependent region.

This concludes the technical overview of IMS/OTMA security options. The next topic presented describes the security capabilities of the IMS Connect program product.

Status of ***Security-Related*** Enhancements Planned For IMS/OTMA

The security-releated enhancements requested by customers for IMS/OTMA are as listed below. An explanation of why the enhancement is needed and the status of the planned delivery are also provided.

1. Improve the performance of security checking for transactions that are set as the destination on CHNG and AUTH calls.

   Explanation: When IMS applications issue either a DL/I CHNG or AUTH call with the destination set to a transaction code, IMS validates that the userid associated with the source transaction is authorized to the transaction set as the destination on the CHNG/AUTH call.

   With the OTMA security level FULL, the ACEE that is built in the control region upon receipt of the input message is propagated to the dependent region and can be used to validate the userid's authority to execute the transaction code requested on the CHNG/AUTH call.

   With the OTMA security level CHECK, however, additional security calls have to be made to build a new ACEE that can be used to validate the userid's authority to execute the transaction code requested on the CHNG/AUTH call. This elongates the time it takes to complete the CHNG/AUTH call, thereby potentially degrading performance. This problem is multiplied many times over for applications that issue many CHNG/AUTH calls and may result in poor performance for CHNG/AUTH calls when the OTMA security level is CHECK.

   Status: The above enhancement is planned for delivery by September 30, 2002, via the service process.

2. Enhance IMS to keep track of previous combinations of userid-transaction verification. This would eliminate (or significantly minimize) userid-to-transaction verifications. IMS/OTMA would not have to reissue the RACROUTE request for the same userid-transaction combination.

   Explanation: When OTMA security is activated, every input transaction request is validated to ensure that the userid associated with the request has access to the transaction. If the same userid sends in multiple requests for the same transaction, each request results in having to issue the same VERIFY and FASTAUTH calls.

   Status: This requirement will not be provided because an enhancement of this nature could introduce security exposures.

   For example, consider the case were a userid (or a group of which a userid is a member) is authorized to execute an IMS transaction. Suppose an installation's security administrator changed the list of authorized userids for a transaction (e.g. removes a userid/group from the transaction's authorized list of userids/groups). If IMS allowed access to the transaction (based on the same userid having previously executed the transaction code) after the security administrator had removed the userid's authority to execute the transaction, a security exposure would be introduced.

   It should also be noted that OTMA's use of a 'hash table' scheme significantly minimizes the number of RACROUTE REQUEST=VERIFY calls already.

3. Enhance the /SECURE OTMA command to support the TMEMBER keyword. This would provide greater granularity on security checking performed for IMS commands and IMS transactions received from a TMEMBER rather than from an OTMA client. As an OTMA client, IMS Connect may communicate with one or more IMS/OTMA datastores and it is desirable to specify security options by individual datastores.

Explanation: The existing /SECURE OTMA command changes the level of security for all OTMA clients. This is a global level and does not take into account that different OTMA clients (e.g. IMS Connect, MQSeries, etc.) oftentimes need different security levels. The result is that the /SECURE OTMA command does not provide the ability to define different OTMA security levels for different OTMA clients.

Status: The above enhancement is planned for delivery by September 30, 2002, via the service process.

4. Increase the size of the OTMA hash tables used for OTMA clients, or provide a mechanism to support a user-defined client hash table size.

   Additionally, provide a 'cast-out' (or ACEE aging) function for specific entries in each client hash table, especially the hash table associated with IMS Connect.

   Explanation: The OTMA hash table entries contain pointers to existing ACEEs for userids that have been previously encountered by OTMA. If an entry for a userid exists in the hash table, the ACEE pointed to by the entry can be reused. This eliminates the overhead of invoking RACF (or an equivalent product) to create an ACEE each time the same userid is encountered by OTMA.

   There is an existing restriction that limits the hash table size to 5,000 entries. Additionally, there is no existing mechanism to request a 'cast-out' of specific entries. This causes the ACEEs for some userids to be reused indefinitely, as is the case for ACEEs created for userids transmitting messages via IMS Connect. These ACEEs may need to be refreshed due to changes made to user and/or group profiles by the installation's security administrators.

   Status: The above enhancement is planned for delivery by December 31, 2002, via the service process.

5. Provide a mechanism for dynamically changing the ACEE aging value associated with a client's hash table on a TMEMBER basis. The ability to change the aging value without recycling OTMA or the OTMA client is also needed.

   Explanation: The ACEE aging value for an ACEE is used to control how long an ACEE is kept in IMS storage for subsequent reuse (i.e. the same userid sends in another transaction request through the OTMA client). The ability to reuse an ACEE, rather than create it for every request, is a performance benefit particularly if the same userid sends in multiple transaction requests. Customers that closely monitor IMS an the associated access from OTMA clients need the ability to request a dynamic override of the ACEE aging value when needed.

   The ability to refresh the ACEE aging value on a TMEMBER basis should be supported on the /SECURE OTMA command to satisfy this requirement.

   Optionally, extend the use of the DFSYDT descriptor to allow specification of an initial ACEE aging value.

   Status: The above enhancement is planned for delivery by December 31, 2002, via the service process.

6. Enhance the Build Security Environment Exit (DFSBSEX0) interface to support transaction input from OTMA and APPC environments. Also, ensure that if the exit exists, it is called for all security levels: NONE, CHECK, FULL, and PROFILE.

   Explanation: Currently OTMA has limited flexibility and control of the security environment for CHNG and AUTH calls when input messages originate from OTMA clients.

   The DFSBSEX0 exit routine can be coded to allow customers to decide on a transaction-by-transaction basis whether transaction authorization is to take place at all for a given transaction code; and if so, when an ACEE is to be built. DFSBSEX0 is driven only when the input is received from non-OTMA and non-APPC environments.

   There is a growing need to provide greater granularity for CHNG/AUTH call security when input messages originate from an OTMA client. For these environments, customers need the ability for the exit, such as DFSBSEX0, to determine the type of security to be performed on a transaction-by-transaction basis rather than on the global OTMA specification (NONE, CHECK, FULL, or PROFILE).

   Status: The above requirement is requires further study and is not planned for immediate delivery.

7. Change the way RACF-protected IMS data set security is implemented for dependent regions in OTMA environments. For example, when a dependent region requests access to an IMS data set.

   a) Allow the ACEE/userid of the dependent region to be used rather than the ACEE/userid associated with the input message when OTMA security level FULL is used.

   b) Allow RACF data set security to be bypassed when the OTMA security level NONE is used.

   Explanation: When the OTMA security level is FULL, if a transaction that originates from an OTMA client accesses a RACF-protected OS data set in a dependent region, the userid used in security checking is that associated with the end user (rather than the userid associated with the dependent region). This requires the security administrator to authorize many different userids/groups access to the data sets. For non-OTMA transactions, the dependent region userid is used thereby limiting the security administration overhead. Increased administrative overhead is encountered when accessing RACF-protected OS data sets when the transaction is entered via an OTMA client. Some customers would like RACF data set authorization checking to be performed using the ACEE associated with the dependent region.

   When the OTMA security level is NONE, if a transaction that originates from an OTMA client accesses a RACF-protected OS data set in a dependent region, RACF is still invoked to perform data set authorization checking. RACF is invoked to perform this authorization even though the OTMA security level has been specified as NONE. Some customers would like RACF data set security to be bypassed when the OTMA security level is NONE.

   Status: The above requirement is requires further study and is not planned for immediate delivery.

Status of ***Non-Security-Related*** Enhancements Planned For IMS/OTMA

1. A.) Enhance OTMA to provide support for OTMA time out similar to the APPCIOT value.

   B.) Additionally, provide the ability to specify a time out value on an inbound commit mode 0 message which could be used to expire or delete the reply if the elapsed time (input receipt to message send) exceeds the time out value.

   Explanation: Installations may experience hung conditions and expired messages due to the limited time out mechanism in OTMA. There are actually two issues involved:

   The first issue has to do with the possibility of a hung OTMA process (e.g. OTMA has sent a message and waits indefinitely for an ACK). There is no time out mechanism or option that allows a timer to pop and force OTMA to clean up the environment.

   The second issue has to do with a message whose processing has been delayed and whose initiating client is no longer available to receive the message. There is no way to specify a time out or expiration value to inform IMS to discard the message if the time has been exceeded.

   Provide the ability to specify a time out value on an inbound commit mode 0 message. This should be used to expire or delete the reply if the elapsed time (input receipt to message send) exceeds the time out value.

   Status: The above enhancement is planned for delivery by December 31, 2002, if time permits, via the service process. Otherwise, the enhancement is planned for delivery as soon as possible in the first half 2003.

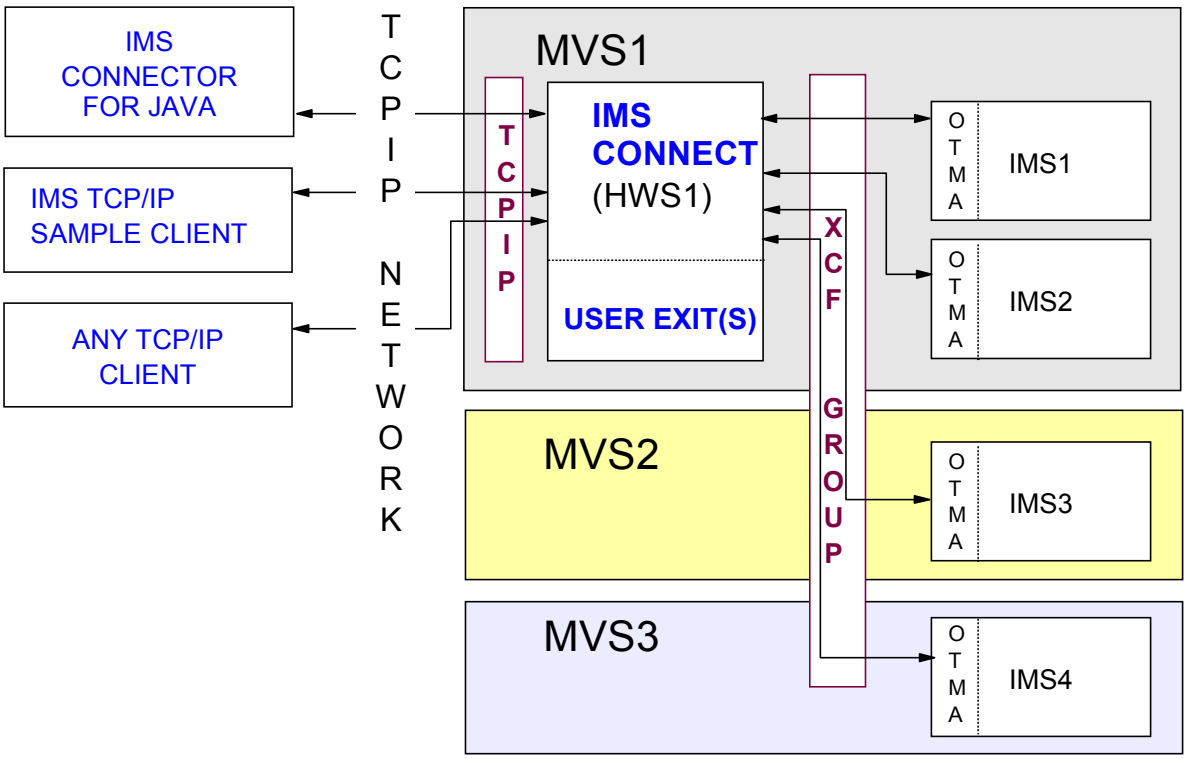
IMS Connect Introduction

IMS Connect is a TCP/IP server that enables TCP/IP clients to exchange messages with IMS OTMA (Open Transaction Manager Access). IMS Connect provides communication linkages between TCP/IP clients and IMS (datastores). It supports multiple TCP/IP clients accessing multiple datastore resources. IMS Connect runs on an MVS, OS/390, or z/OS platform.

IMS Connect performs router functions between TCP/IP clients and datastores. Input (request) messages received from TCP/IP clients, via TCP/IP connections, are passed to an IMS datastore through XCF sessions. IMS Connect receives response messages from the datastore and then passes them back to the originating TCP/IP clients.

IMS Connect supports TCP/IP clients communicating with socket calls, but it can support any TCP/IP client that communicates with a different input data stream format. User-written message exits can execute in the IMS Connect address space to convert customer message format to OTMA message format before IMS Connect sends the message to IMS. The user-written message exits also convert OTMA message format to customer message format before sending a message back to IMS Connect. IMS Connect then sends output to the client. Refer to *Figure 11 IMS Connect.*

FIGURE 11. IMS Connect

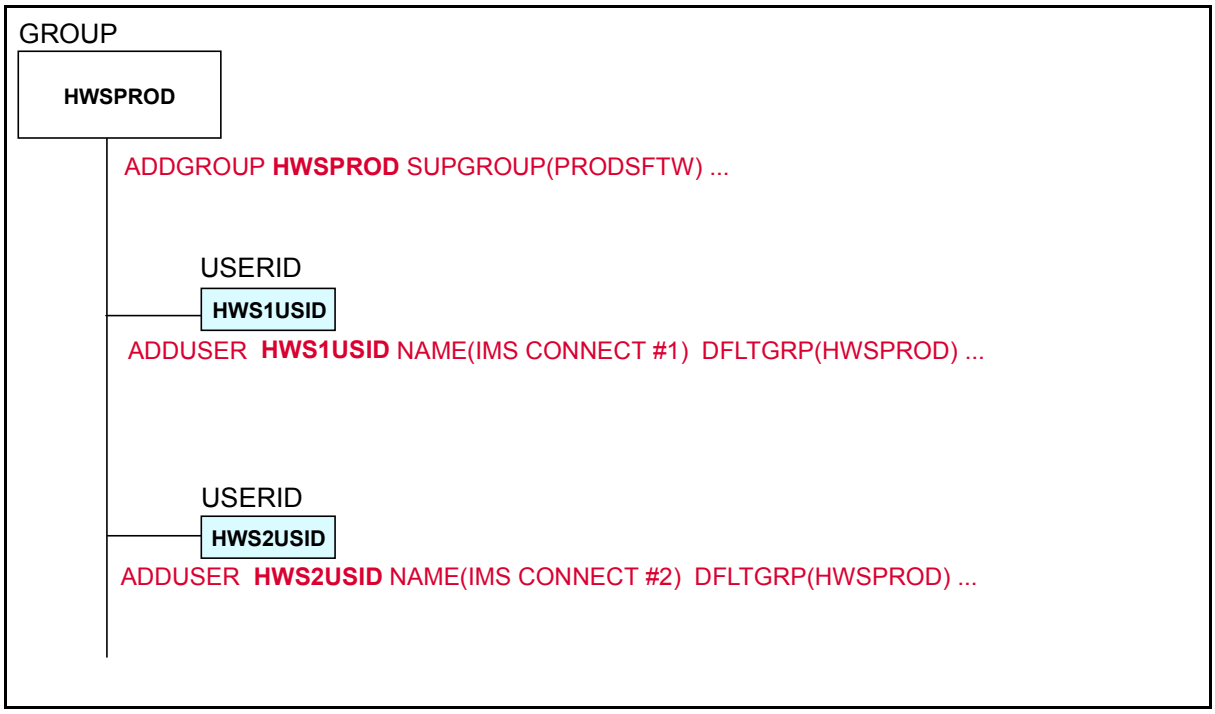


In addition to TCP/IP client communications, IMS Connect also supports local communications through the "Local" option. This option provides a non-socket (non-TCP/IP) communications protocol for use between IBM WebSphere and IMS Connect in OS/390 and z/OS operating system environments. Using the local option, IMS Connect can communicate with servlets executing in IBM WebSphere and using IMS Connector for Java.

Supplying the Userid and Group For the IMS Connect Product

Like other products, IMS Connect must be assigned a RACF userid and connected to a RACF group. The userid and group for IMS Connect will be supplied in the client-bid message when IMS Connect attempts to connect to IMS/OTMA. If IMS/OTMA security is activated (turned on), the IMS Connect userid (and optionally, the group name) will be required for a successful client-bid.

The RACF security administrator can create a userid for IMS Connect by issuing the RACF ADDUSER (AU) command. The userid for IMS Connect can be connected to an existing RACF group; or alternatively, the administrator can create a new RACF group for IMS Connect by issuing the RACF ADDGROUP (AG) command. See the diagram below for an illustration of the ADDUSER and ADDGROUP commands.



If the datastore (which is IMS) is RACF protected, you can start IMS Connect as a job with the JOB card specifying a valid USERID in order to make the connection from IMS Connect to IMS. Or alternatively, you can use the RACF STARTED class or the Started Procedure Table to associate the IMS Connect userid to the started procedure used to initialize IMS Connect.

The USERID=&userid parameter specified in the JOB card of the IMS Connect job JCL, or the userid association made in the STARTED class and/or Started Procedure Table, is used as the security vehicle to ensure IMS Connect access to IMS. The IMS Connect userid must have at least READ access to FACILITY class profile used to secure the client-bid connection request. FACILITY class profiles used to secure the client-bid connection requests use a naming convention of the format: IMSXCF is always the high level qualifier; the XCF group name that both the OTMA client (i.e. IMS Connect) and IMS/OTMA joined is the middle level qualifier; and the XCF member name for the OTMA client (which in this case is IMS Connect) is the low level qualifier. See the diagram below for an illustration of the profile naming convention.

You also need to update the MVS Program Properties Table (PPT) to allow IMS Connect to run in authorized supervisor state and in Key 7 storage.

```
PPT PGMNAME(HWSHWS00)    /* PROGRAM NAME = HWSHWS00                          */
KEY(7)                   /* PROTECT KEY ASSIGNED IS 7                          */
PASS                     /* CANNOT BYPASS DATASET PASSWORD PROTECTION  */
SYST                     /* PROGRAM IS A SYSTEM TASK                         */
...
```

MVS PPT

This concludes the information presented for the IMS Connect product itself. The remainder of the information regarding IMS Connect security pertains to security checking performed by IMS Connect for the TCP/IP end users or TCP/IP client applications.

End User Validation and Verification Performed By IMS Connect

IMS Connect allows security support by checking a RACF flag. If you turn the RACF flag ON, IMS Connect uses the System Authorization Facility (SAF) interface to invoke RACF by issuing a RACROUTE REQUEST=VERIFY call to verify a userid and password. This is a global option in that IMS Connect either performs verification for all users or does not perform verification for any users. There are two ways to activate the RACF flag:

- Set the RACF flag in the configuration file, HWSCFG00, by setting the flag to Y (RACF=Y).

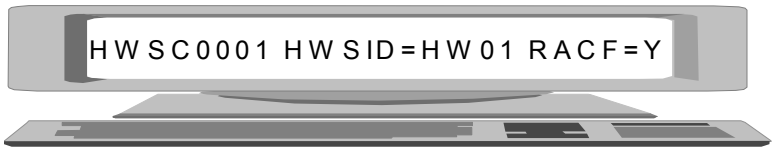
  HWS ID=HWS01 RACF=Y

  HWSCFGxx

  HWS (ID=HWS01,**RACF=Y**)
  TCPIP (...**RACFID=*default_userid*,EXIT=(HWSIMSO0,HWSJAVA0,**...)
  DATASTORE (ID=IMS1,GROUP=**XCFGRP1**,MEMBER=**HWSMEM**,TMEMBER=**IMS1MEM**,DRU=HWSYDRU0)
  DATASTORE (ID=IMS2,GROUP=**XCFGRP1**,MEMBER=**HWSMEM1**,TMEMBER=**IMS2MEM**,DRU=HWSYDRU0)
  ...

- Issue the HWS command **SETRACF** to set the RACF flag.

To check the setting of the RACF flag, you can issue the **VIEW HWS** command. After you issue this command, you should see:



The security information pertaining to end users must be passed to IMS Connect in order for IMS Connect to invoke RACF to verify userids associated with end users. Security information is passed from TCP/IP clients in the IMS Request Message (IRM). See the illustration below to see the portion of the IRM that has the security-related information.

```
IRM
.
.
.
 IRM_RACF_USERID      8 BYTES     RACF USERID PROVIDED BY:
                                        TCP/IP CLIENT, OR
                                        IMS CONNECT EXIT IN OTMA HEADER FIELD OMSECUID
 IRM_RACF_GRNAME      8 BYTES     RACF GROUP NAME PROVIDED BY:
                                        TCP/IP CLIENT, OR
                                        IMS CONNECT EXIT IN OTMA HEADER FIELD OMSECGRP
 IRM_RACF_PW          8 BYTES     RACF PASSWORD OR PASSTICKET PROVIDED BY:
                                        TCP/IP CLIENT
 .
 .
 .
```

The userid, RACF group name (optional), and password in the IRM may originate from one of the following sources:

- The **end user** at the terminal.
- A **security exit** routine invoked from the message exit used to perform message translation.
- The **default userid** supplied by the IMS Connect startup/execution parameter **RACFID=**.

In order for the userid and password supplied by the end user and/or the security exit to be used in VERIFY security checking, the IRM must include definition fields for the userid and password and optionally, a RACF group name. Also the values in the respective fields can not be nulls or blanks. **'Appendix D. HWSSMPL0, HWSSMPL1, HWSIMSO0, and HWSIMSO1 Security Actions**' in the ***'IMS Connect Guide and Reference'*** manual provides tables that show the userid, password, and group name used in VERIFY processing when a security exit ***is not*** invoked versus when a security exit ***is*** invoked. These tables also represent the userid and optionally, group name, that will be passed to IMS/OTMA in each message.

Currently, if configured to perform userid verification security checking, IMS Connect verifies ***BOTH*** the userid ***and*** password.

IMS Connect User Message Exit Routine

Each message received by IMS Connect must be processed by a user message exit routine in order to translate/format the message so that it is in a format acceptable to IMS/OTMA. The user message exit invoked to process a message depends on the source/origination of the message. For example, if the IMS Connector for JAVA client application is the source of a message, the user message invoked to format the message is HWSJAVA0. Conversely, if the source of a message is ***not*** IMS Connector for JAVA, the installation could use either the HWSSMPL0 or HWSSMPL1 user message exit to translate the message.

IMS Connect supports the capability to invoke a security exit routine from the message exit routines that are used to translate messages. In some cases, the installation may require more extensive security than userid and password verification checking performed by IMS Connect. For example, the installation may want to do one or more of the following:

- Perform some security checking based on the client's (e.g. IMS Connector for JAVA) IP address and port number.
- Perform **userid validation only**; that is, perform userid VERIFY security without password verification for the userid.
- Use the APPL operand on the RACROUTE REQUEST=VERIFY macro issued from a security exit routine and RACF APPL class profiles to control which users can use IMS Connect as they enter the system.
- And so forth.

User Security Exit Routine

Either a user-provided security exit routine or the IMSLSECX sample security exit provided by TCP/IP may be customized to meet installation security requirements. There are some security exit routine naming standards that you should be familiar with prior to invoking a security exit from a message exit:

- If a security exit routine is invoked from either the **HWSIMSO0** or **HWSIMSO1** user message exit routine, the security exit name ***must be*** **IMSLSECX**.
- If a security exit routine is invoked from either the **HWSSMPL0** or **HWSSMPL1** user message exit routine, the security exit name may be determined by the installation. Or alternatively, the installation may invoke the IMSLSECX security exit routine.
- If a security exit routine is invoked from the **HWSJAVA0** user message exit routine, the security exit name may be determined by the installation. Or alternatively, the installation may invoke the IMSLSECX security exit routine.

The IMSLSECX security exit routine and user-code security exit routines are described in Chapter 5, 'IMS Connect User Message Exit Support', in the ***IMS Connect Guide and Reference*** manual.

User exit routines that are to be invoked by IMS Connect are specified on the EXIT= keyword of the TCPIP statement in the HWSCFG file. Currently a maximum of 15 exits may be specified on the EXIT= keyword. Refer to **TCPIP statement** in the HWSCFG diagram in the topic '*End User Validation and Verification Performed By IMS Connect'* for an illustration of how to code the EXIT= keyword.

Refreshing Security Control Blocks in IMS/OTMA

When IMS Connect establishes communications with IMS/OTMA, IMS Connect indicates on the client-bid message how long IMS/OTMA should retain ACEEs that were created as a result of receiving input messages from IMS Connect users. As you recall, the ***ACEE aging value*** in the State Data (SD) portion of the client-bid message specifies how long IMS/OTMA can use ACEEs created as a result of receiving end user messages from IMS Connect.

Currently, the ACEE aging value supplied by IMS Connect on the client-bid message is the default of 2,147,483,647 seconds. For messages received by IMS/OTMA from IMS Connect, the default value (2,147,483,647 seconds) is fixed and can not be changed by the installation. What the aging value does is determine the length of time, in seconds, that a userid from IMS Connect is considered previously verified by IMS/OTMA. This amount of time may be too long for some IMS installations. To cause hash table for IMS Connect to be refreshed in IMS/OTMA and subsequently refresh ACEEs created in IMS/OTMA for userids associated with messages received from IMS Connect, take the following steps:

- Issue the STOPDS command to IMS Connect. The STOPDS command immediately terminates communication between the IMS Connect and a datastore. IMS/OTMA deletes the hash table for IMS Connect when the communication is terminated.

  For example, to stop IMS Connect communications with IMSA, enter the following command: *nn*STOPDS IMSA, where *nn* is the reply number of the outstanding reply message and IMSA is the value supplied on the ID= keyword on the DATASTORE statement in the HWSCFG file containing the IMS Connect startup parameters.

- Issue the OPENDS command (e.g. *nn*OPENDS IMSA) to reestablish communication between the IMS Connect and IMS/OTMA. IMS/OTMA builds an empty hash table for IMS Connect upon a successful client-bid.

A nice feature of IMS Connect is the ability to issue the above commands to stop and start communications with an IMS datastore. These commands allow the installation to refresh the security information in the hash table used by IMS/OTMA (for messages from IMS Connect) **without having to stop and restart OTMA**. Some OTMA clients, which require you to stop and

restart OTMA in order to refresh the IMS/OTMA's hash table for the OTMA client. This disrupts all OTMA processing and causes all hash tables to be deleted and rebuilt for all OTMA clients.

Status of ***Security-Related*** Enhancements Planned For IMS Connect

- The security-related enhancements planned for ***IMS Connect*** are as follows:

1. Provide an 'already verified' flag in the message header (for example, the IRM) so that IMS Connect can issue a RACROUTE REQUEST=VERIFY,PASSCHK=NO call. This support would provide a bypass for the password check, thereby causing RACF to validate a userid without verifying the password for that userid.

   Explanation: The existing security implementation of IMS Connect assumes that if security is to be invoked then a valid password must always accompany a userid. The client-server world, however, has defined the concept of 'trusted users'. One definition of this concept allows for end users to be validated and authenticated at a remote security server and subsequent access to a back-end host or server to pass the userid with a flag indicating ' already verified'. This allows validation of the userid on the back-end host/server to occur without making a secondary authentication for password.

   This capability also addresses the issue of an expired password because, in this case, the password is not checked.

   Status: The above enhancement is planned for delivery by June 30, 2002, via the service process.

2. Provide the ability to reuse ACEEs and/or implement a hash table scheme along with an ACEE aging value similar to that used by MQSeries and IMS/OTMA.

   Explanation: When RACF=Y is specified in the configuration file, IMS Connect issues a RACROUTE call to the underlying security product for each input message that carries a security header. Part of this process includes creating an ACEE that is associated with the userid. Since it is common for the same userid to send in multiple transactions, the overhead of requesting the creation of an ACEE multiple times for the same userid can be unnecessary overhead. This is a consideration in both the use of persistent and/or transaction sockets from the same user.

   Status: The above enhancement is already available through the RACF Virtual Lookaside Facility (VLF) ACEE caching mechanism. The VLF ACEE caching facility is deemed to be more appropriate and may be used in lieu of a 'hash table' scheme implementation in IMS Connect.

   Upon receipt of a message, IMS Connect issues a RACROUTE REQUEST=VERIFY call to RACF (or equivalent) to verify the userid and password in the incoming message. If the installation has implemented VLF ACEE caching, the ACEE for the userid in VLF is used, thus eliminating I/O to the RACF database when the ACEE is already in the cache.

3. Provide more options to control security.

   Explanation: Installations have a limited capability to control the security environment. The current security specifications are determined globally for the IMS Connect system by specifying RACF= on the HWS statement and optionally RACFID= on the TCPIP statement in the HWSCFG file. This does not allow the flexibility to request different security options on a more granular level. For example:

   - Need the capability to bypass the security check for certain transactions even though RACF=Y has been specified (on the HWS statement in the HWSCFG configuration member).
   - Need the IMSLSECX security exit to be enhanced to have greater flexibility and control.
   - Allow the specification of the RACF= and RACFID= keywords on the DATASTORE statement to override the global values provided on the HWS and TCPIP statements (in the HWSCFG configuration member).

   Status: The above enhancements will be delivered by allowing the user message exit routine to override the RACF=Y option. The enhancement is planned for delivery by June 30, 2002, via the service process.

4. Enhance IMS Connect to provide support for a 'password change' and a 'new password reverify' function.

   Explanation: When a password for a userid expires, the RACROUTE authentication call fails and a security failure error indicator is sent back to the IMS Connect remote client. There is no support in place in IMS Connect that facilitates the ability to change passwords. The existing message header (IRM) format does not provide a field to enter a new password (in addition to the old); nor does IMS Connect have the supporting code to request a password change or to reverify a changed password.

   Status: The above requirement may be met by the 'trusted user' support described in #1 above. The IRM format will be enhanced to allow the IMS Connect client to set an 'already verified' flag to indicate that only the userid is to be verified (no password authentication). When IMS Connect detects that the 'already verified' flag has been set in an incoming message, IMS Connect will invoke RACF with a PASSCHK=NO specification on the RACROUTE REQUEST=VERIFY macro.

5. Externalize the ACEE aging value in a parameter of the HWSCFG configuration member. If the value is not defined, then default to the existing maximum specification.

   Explanation: IMS Connect has a hard coded specification of 2,147,483,647 seconds for the ACEE aging value. There is currently no way to specify an override for this value. Other OTMA clients, such as the MQSeries-IMS Bridge application, provide the ability to specify an override.

   Status: The above requirement is planned for delivery by December 31, 2002, via the service process.

6. Enhance IMS Connect to support Secure Socket Layer (SSL), PassTickets, and digital certificates.

   Explanation: Although IMS Connect provides the option of calling an underlying security product to validate and authenticate userid/passwords associated with inbound messages, there exist requirements for other security mechanisms that are part of the evolving Internet environment.

   Any enhancement in this arena should also take into consideration that IMS Connect, actually multiple IMS Connect subsystems, could be part of a Sysplex environment that supports some kind of TCP/IP generic routing mechanism. An example of the impact of this configuration, for example, would impact PassTicket

   support. For PassTickets, IMS Connect would have to issue a RACROUTE call that would specify the target APPL (i.e. the IMS Connect name). Since multiple IMS Connect subsystems could be part of the generic environment, then a mechanism should be implemented that would allow all the IMS Connect subsystems to specify the same APPL name. IMS already does something similar to this with the SAPPLID= startup parameter that was introduced in the IMS V7 PassTicket support.

   Status: The above requirement is planned for delivery by December 31, 2002, via the service process.

7. Allow all message exits that are shipped with IMS Connect and associated connector products to invoke user-written routines.

   Explanation: There are inconsistent security capabilities from message exit routines. HWSJAVA does not call the security exit. All message exits should be able to access user-written routines that implement additional user-specific security checks.

   Status: The above requirement is planned for delivery by June 30, 2002, via the service process.

Status of Non-security-related Enhancements Planned For IMS Connect

The non-security-related enhancements planned for IMS Connect are listed below in bold text. A brief explanation of why the enhancement is needed is also provided. Finally, after each planned enhancement, the status of the planned enhancement is provided.

1. Provide a time out field in the message header format (for example, the IRM) that overrides the global value in the HWSCFG file for a specific message.

Explanation: IMS Connect provides a global time out value in the TCPIP statement of the HWSCFG file. This value prevents the IMS Connect client from appearing to be 'hung' and determines the amount of time that IMS Connect is to wait for either a response from:

a.) IMS to send to the client, or

b.) The client to send to IMS.

There is a growing customer need to make this capability more granular so that different time out levels can be specified on a message basis.

Status: The above enhancement is planned for delivery by June 30, 2002, via the service process.

2. Increase the number of user exits that can be specified from 15 to 256.

Explanation: There is a current maximum for the number of exits that can be specified in the TCPIP statement of the HWSCFG file. This is a severe restriction for some customer environments and can result in having to initialize more than one IMS Connect address space just to support the exits.

Status: The above enhancement is planned for delivery by June 30, 2002, via the service process.

3. Enhance IMS Connect to ***automatically*** reconnect to a datastore (IMS) when IMS has been restarted..

Explanation: During initialization, IMS Connect attempts to establish connectivity with all defined DATASTOREs, (e.g. IMS systems). If, however, IMS is not active at this time or is recycled during the lifetime of an IMS Connect system, no automatic reconnect is attempted when IMS becomes available. The

OPENDS command has to be issued for the connection to deactivated. Customers have had to do this manually or by an automated operations process. Since an XCF signal is always sent to IMS Connect whenever an IMS in the XCF group is activated, IMS Connect should react to this and automatically reconnect.

Status: The above enhancement is planned for delivery by June 30, 2002, via the service process.

4. Enhance IMS Connect to support distributed commit (Commit mode 2).

Explanation: Client programs today have the choice of specifying either Commit-then-Send (Commit mode 0) or Send-then-Commit (Commit mode 1) to control when IMS sends the output reply relative to IMS sync point. There is an evolving requirement, however, to allow the client program to have greater control of the commit scope. Specifically, one that would allow a coordinated commit for multiple resources where IMS is possibly just one of many resources. This capability is already supported for applications that use APPC to communicate with IMS an is also supported by OTMA CI (callable interface) and the OTMA component in IMS. IMS Connect needs to take advantage of this capability.

Status: A date has *not* been established for delivery of this planned enhancement.

5. Provide a way for IMS Connect to automatically reroute a message to an alternate IMS if the target system is unavailable or undefined. The search order for active DATASTOREs should be specified in either:

a) The DATASTORE/INIT tables or

b) Through a new statement in the HWSCFG configuration member

Explanation: Although IMS Connect has provided a DATASTORE table and an exit interface that can access the table to determine the status (active or inactive) of an IMS system, the ability to take advantage of this functionality requires the customer to write code in the exits. Many customers feel that IMS Connect should be able to provide this functionality without requiring customer-written code.

Status: A date has *not* been established for delivery of this planned enhancement.

6. Provide IMS Connect support for Open Database Access (ODBA) environments.

   Explanation: There is a requirement for direct access to IMS databases from web server programs. Today, access to IMS through IMS Connect assumes connectivity to IMS/TM for access to transactions and commands. There is a growing need to access IMS databases directly from Web server programs without invoking an IMS transaction.

   Status: A date has *not* been established for delivery of this planned enhancement.

Technical Overview of MQSeries-IMS Bridge Security

Most businesses have networks of diverse hardware interoperability and software. However, related programs in different parts of a network must be able to communicate in a way unaffected by variations in hardware, in operating systems, in programming languages, and in communication protocols. Moreover, businesses need to be able to run related programs independently of each other. And all this needs to be achieved with an overall reduction in the number of sessions on the network.

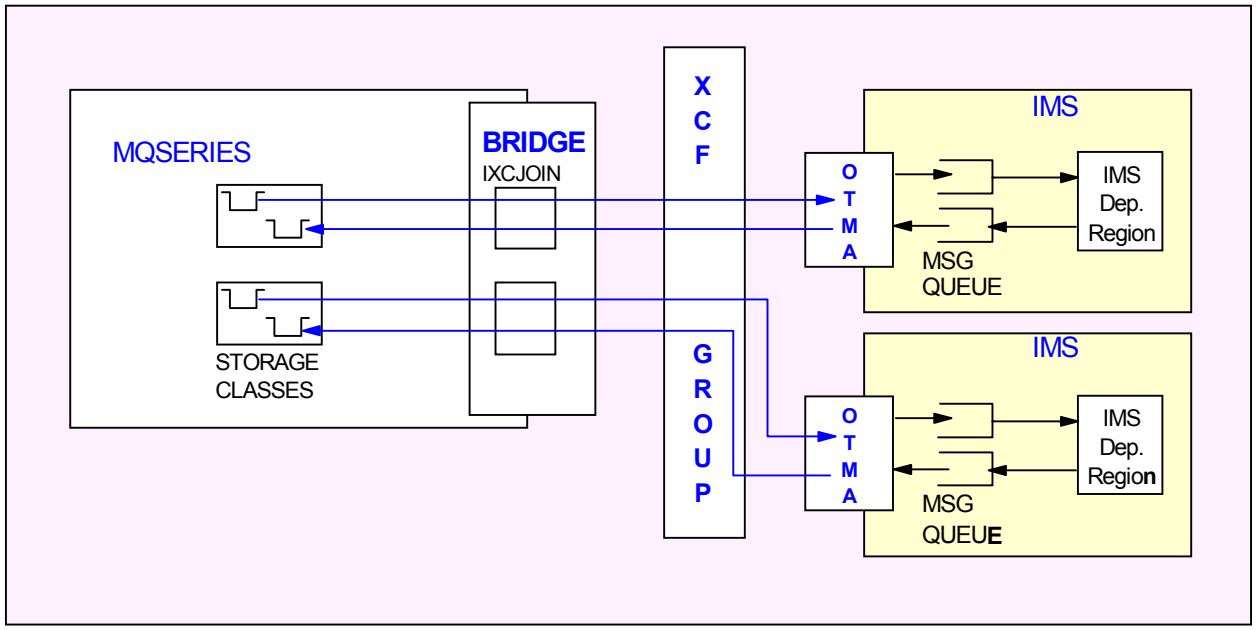
Complex though the problem may be, it needs a solution that works in the same way (and equally well) between programs on a single processor (in both like and unlike environments) and between programs at different nodes of a varied network.

MQSeries products provide just such a solution. MQSeries enables programs to communicate across a network, without having a private, dedicated, logical connection to link them. And it does this in a way that's simple, elegant, and proven: programs communicate by putting messages on message queues, and by taking messages from message queues.

The MQSeries-IMS Bridge is the **component of MQSeries** for OS/390 that allows direct access from MQSeries applications to applications on your IMS system (the bridge enables implicit MQI support). This means that you can re-engineer legacy applications that were controlled by 3270-connected terminals to be controlled by MQSeries messages, without having to rewrite, recompile, or relink them. The Bridge is an IMS Open Transaction Manager Access (OTMA) client.

In Bridge applications there are no MQSeries calls within the IMS application. The application gets its input using a GET UNIQUE (GU) to the IOPCB and sends its output using an ISRT to the IOPCB. MQSeries applications use the IMS header (the MQIIH structure) in the message data to ensure that the applications can execute as they did when driven by non-programmable terminals. If you are using an IMS application that processes multi-segment messages, note that all segments should be contained within one MQSeries message. The MQSeries-IMS Bridge application is illustrated in *Figure12. MQSeries-IMS Bridge and IMS/OTMA Communications*.

FIGURE 12. MQSeries-IMS Bridge and IMS/OTMA Communications



Implementing MQSeries-IMS Bridge Security

There are several aspects that you must consider for security for MQSeries-IMS Bridge application, which are:

- Defining the MQSeries security options to take place for messages destined for IMS/OTMA.
- Configuring MQSeries to join the same XCF group as the IMS/OTMA system with which the MQSeries-IMS Bridge application will communicate.
- Configuring IMS to join the same XCF group as the MQSeries which will pass messages to IMS/OTMA via the MQSeries-IMS Bridge application.
- Defining the MQSeries objects: storage class, Bridge queue used to transmit messages to IMS/OTMA, and the reply queue used to receive reply messages from IMS/OTMA.
- Operating the MQSeries-IMS Bridge, and

Defining the MQSeries Security Options (for IMS/OTMA Messages)

Before you attempt to transmit messages (client-bid **and** end user messages) to IMS using the MQSeries-IMS Bridge, you should determine how much userid verification checking you want MQSeries to perform for messages destined for IMS/OTMA. The amount of userid verification you want MQSeries to perform is documented in:

> *MQSeries for OS/390 VxRx System Management Guide* manual (SC34-5374)
> Chapter title: *Security*
> Section title: *MQSeries Security Implementation*
> Topic title: *Security Considerations For Using MQSeries with IMS*
> **Sub topic title: *'Security Considerations For the IMS Bridge'*** (NOTE: The most important information is in the subject titled ***'Application Access Control'***.

During MQSeries initialization, MQSeries checks its startup/execution parameters and determines whether or not the MQSeries-IMS Bridge application will be used to transmit messages to IMS/OTMA. If MQSeries detects that the MQSeries-IMS Bridge will be used, MQSeries invokes RACF (or equivalent) to:

- Determine if a profile exist in the FACILITY class to inform MQSeries how much (if any) userid verification should be done for messages destined for IMS/OTMA.
- Return (to MQSeries) the access level associated with the MQSeries userid in the FACILITY class profile.

MQSeries uses the information extracted from RACF to establish the level of userid validation checking and UTOKEN caching will take place in MQSeries for messages destined for IMS/OTMA.

FACILITY Class Profile

When MQSeries invokes RACF to check for a FACILITY class profile and the MQSeries userid access level on that profile, a profile naming convention is used. The naming convention is as follows: IMSXCF.*XCF_group_name*.*XCF_member_name_for_IMS*, where:

- IMSXCF is always the high level qualifier in the profile name.
- The second qualifier in the profile name is the name of the XCF group that both the MQSeries-IMS Bridge application and IMS/OTMA join in order to communicate using XCF.
- The third level (or low level) qualifier in the profile name is the XCF member name for the **IMS**/OTMA system.

See the diagram below for an example of a FACILITY class profile that uses the naming convention described above.

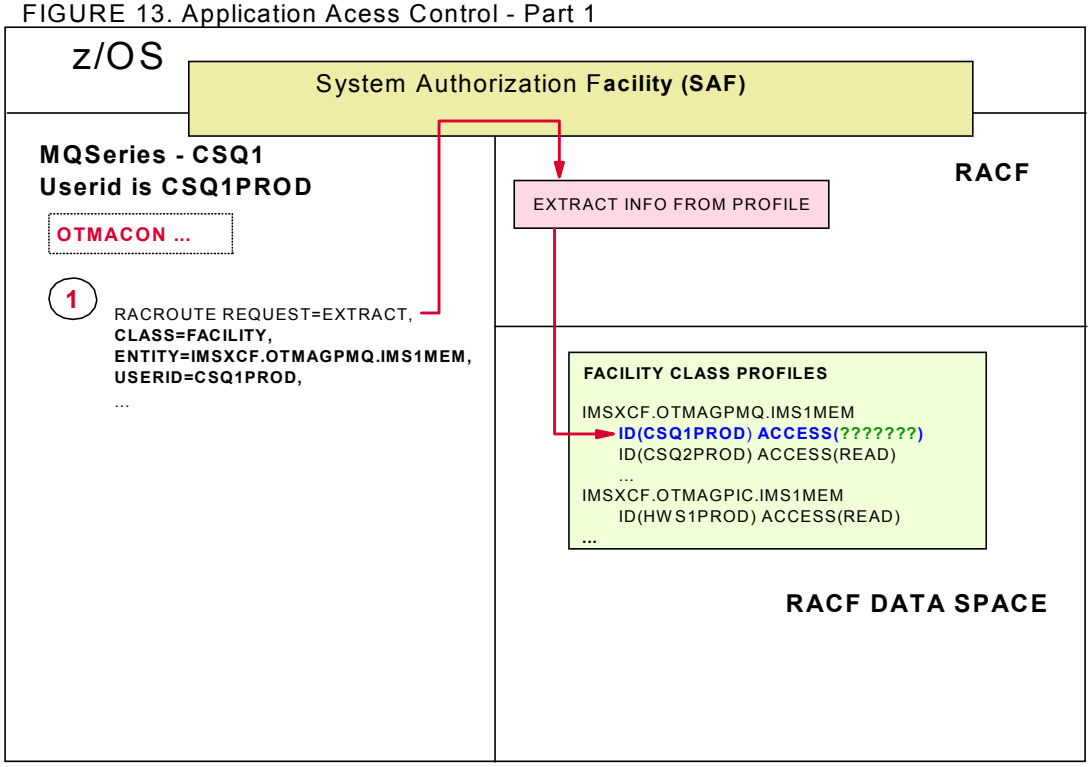
IMSXCF.OTMAGPMO.IMS1MEM

A FACILITY Class profile with a name of 'IMSXCF.OTMAGPMQ.IMS1MEM' means the following:

- **IMSXCF** is the high level qualifier in the profile name.
- **OTMAGPMQ**, the second level qualifier in the profile name, is the **XCF group name** which both the MQSeries Bridge application and IMS/OTMA join at startup. Both MQSeries and IMS would supply this XCF group name in startup/execution parameters.
- **IMS1MEM**, the third and low level qualifier in the profile name, is the **XCF member name for IMS.** The XCF member name for IMS is supplied by **IMS** on the **OTMANM=** keyword in startup/execution parameters in the DFSPBxxx member of IMS.PROCLIB, for example: OTMANM=IMS1MEM. If the OTMANM= keyword is not present in IMS startup parameters or if a value is not supplied on the OTMANM= keyword, IMS uses a default. The default value is the ***value*** supplied on the **APPLID=** keyword on the **COMM macro** used to generate the IMS subsystem. The value supplied on the APPLID= keyword may be overridden by specifying a value on the **APPLID1=** keyword in IMS.PROCLIB(**DFSPBxxx**).

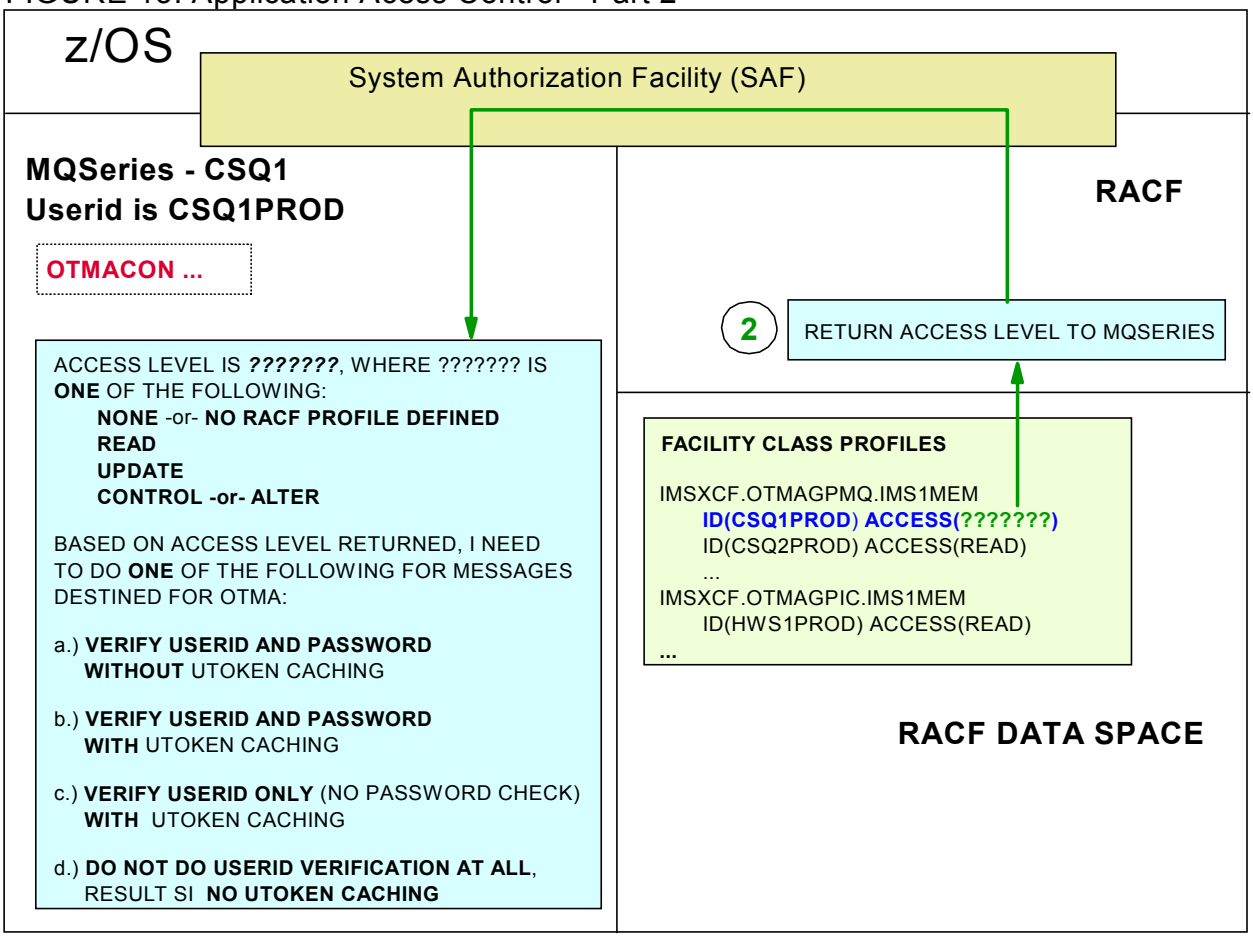
It should become clearer in a while how MQSeries uses the access level information from the profile. For now, refer to *Figure 13, Parts 1 and 2, 'MQSeries Application Access Control'*. Note that MQSeries invokes RACF to extract info from the profile.

FIGURE 13. Application Acess Control - Part 1



During MQSeries initialization, MQSeries notes the presence of the OTMACON execution/startup parameter. The presence of the OTMACON parameter denotes that the MQSeries-IMS Bridge application will be used to transmit messages to IMS/OTMA via XCF. MQSeries needs to determine how much userid validation and UTOKEN caching is needed for messages destined for IMS/OTMA. Hence, MQSeries issue a RACROUTE call to RACF to retrieve information stored in a FACILITY Class profile. The information retrieved by RACF is the access level specified for the MQSeries userid on a profile of the form IMSXCF.*XCF_group_name*.*XCF_member_name_for_IMS* . In Figure 13 - Part 1, the name of the profile is IMSXCF.OTMAGPMQ.IMS1MEM.

FIGURE 13. Application Acess Control - Part 2



The information returned by RACF helps MQSeries decide on the amount of userid validation and UTOKEN caching to perform for messages destined for IMS/OTMA via the Bridge. As Figure 13 - Part 2 indicates, RACF returns one access level (NONE, READ, UPDATE, CONTROL, or ALTER) to MQSeries or RACF returns a 'no profile defined' return code to MQSeries. The following paragraphs and figures describe the userid verification performed by MQSeries when each of the access levels (or 'no profile defined' condition) is returned. ***Your RACF security administrators should consider the implications of each access level before defining the FACILITY Class profile and authorizing the MQSeries userid with one of the access levels***.

ID(*MQ_userid*) ACCESS(NONE) or NO PROFILE DEFINED

When MQSeries is returned an ACCESS(NONE) access level or a 'no profile defined' return code, this indicates that the maximum level of userid verification checking is required for each message that the MQSeries-IMS Bridge application transmits to IMS/OTMA. MQSeries invokes RACF to verify all of the following:

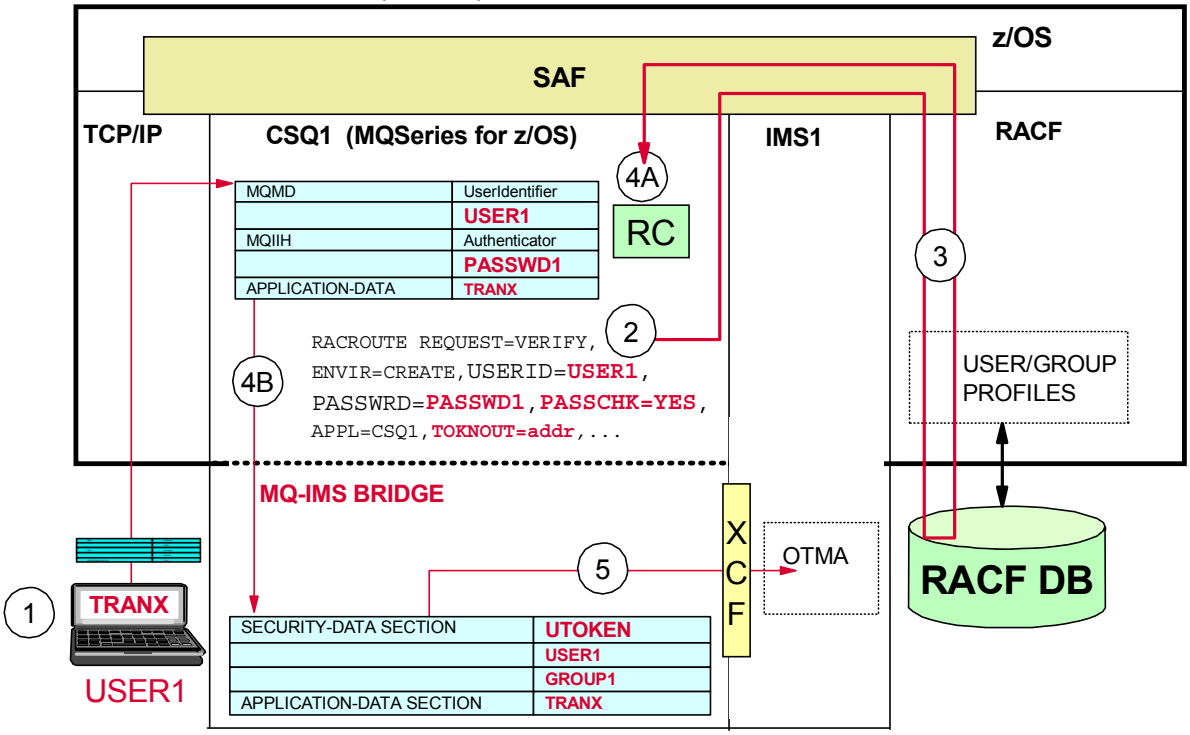
- The userid specified in the UserIdentifier field of the MQMD structure in each message destined for IMS/OTMA via the MQSeries-IMS Bridge has been defined to RACF.
- The password or PassTicket in the Authenticator field of the MQIIH structure in each message destined for IMS/OTMA via the MQSeries-IMS Bridge is also known to RACF.
- The userid in the UserIdentifier field and password/PassTicket in the Authenticator field are a valid combination.

When ACCESS(NONE) or 'no profile defined' is returned to MQSeries:

- MQSeries will invoke RACF to verify both the userid and password (or PassTicket) each time a message is received that is destined for IMS/OTMA.
- A UTOKEN (user token security control block) is returned to MQSeries for verified userids and it is passed to IMS in the message.
- The UTOKEN is not cached in MQSeries controlled storage.

Figure 14. ACCESS(NONE) or NO PROFILE DEFINED illustrates verification performed by MQSeries and RACF (or an equivalent security product).

FIGURE 14. ACCESS(NONE) or NO PROFILE DEFINED



At **step** 1 the userid, USER1, enters a message that request the execution of the IMS transaction code TRANX. Upon receipt of the message (at **step 2**) MQSeries for z/OS invokes RACF by issuing a RACROUTE REQUEST=VERIFY call. Note that MQSeries supplies all of the following information on the RACROUTE request macro:

- The userid (USER1).
- The password or PassTicket (PASSWD1).
- PASSCHK=YES which indicates that RACF is to verify the password or PassTicket for the userid.
- TOKNOUT= keyword which indicates that RACF is to return a UTOKEN if the userid and password/PassTicket are a valid combination.

At **step 3**, RACF checks the user and group profiles to determine if the userid (USER1) has been defined to RACF and that the password/PassTicket (PASSWD1) is valid for the userid. RACF returns a return code to MQSeries indicating the result of the verification check. If the RACF check was successful (return code 0) a UTOKEN is also returned to MQSeries. At **Step 4A**, MQSeries receives the RACF return code (RC). If the userid/password or PassTicket was a valid combination, at **step 4B** MQSeries places the UTOKEN in the message and passes the message to the MQSeries-IMS Bridge application. The MQSeries-IMS Bridge application performs the message translation. That is, the Bridge formats the message in a manner acceptable to IMS/OTMA.

After the message is properly translated by the MQSeries-IMS Bridge application, as **step 5** shows, the message is passed to IMS/OTMA. As you may recall from Figure 6 (steps 3 and 3A) in this document, IMS/OTMA will supply the

UTOKEN address on a RACROUTE macro when IMS invokes RACF to build an Accessor Environment Element (ACEE) in the IMS control region for the previously verified userid (USER1). Use of the UTOKEN improves IMS-RACF verify processing.

Finally, you should note that if profile **qmgr.NO.SUBSYS.SECURITY** (where *qmgr* is the subsystem identifier for the MQSeries queue manager) exists in the **MQADMIN Class**, this level of security on this profile overrides whatever the ACCESS level is defined in the FACILITY Class IMSXCF.*XCF_group.name.XCF_member_name_for_IMS* profile.

ID(*MQ_userid*) ACCESS(READ)

When MQSeries is returned an ACCESS(READ) access level, this indicates that the maximum level of verification checking is also required for each message that the MQSeries-IMS Bridge application transmits to IMS/OTMA. The difference between ACCESS(NONE) and ACCESS(READ) is that MQSeries uses a UTOKEN caching scheme when the ACCESS(READ) level is returned by RACF. Remember UTOKEN caching is not performed by MQSeries when the ACCESS(NONE) is returned.

When the ACCESS(READ) level is returned, MQSeries does one or more of the following:

- ***Only*** issues a RACROUTE REQUEST=VERIFY call to RACF to verify that the userid in the incoming message is valid and the password/PassTicket is also valid for the userid when it is necessary.
  - If this is the very first time MQSeries has encountered a userid in an incoming message, invoke RACF to perform verification.
  - When MQSeries has previously encountered the userid in a prior message, but the UTOKEN in the MQSeries UTOKEN cache has expired (exceed the amount of time MQSeries considers as a valid UTOKEN), invoke RACF to perform verification.

    The amount of time that MQSeries considers a UTOKEN in the cache to be valid is set by the MQSeries command: ALTER SECURITY INTERVAL(*integer*) TIMEOUT(*integer*)**.** The TIMEOUT integer value is the time period in minutes that an unused userid can remain signed on within the MQSeries subsystem. The INTERVAL integer value is the time period in minutes between MQSeries checks for userids for which the TIMEOUT has expired. For example, if the TIMEOUT value is 30 minutes and the INTERVAL value is 10 minutes; every 10 minutes MQSeries checks for userids that have not been used for 30 minutes. If such a userid is found, that userid is signed off within the queue manager.

    When the TIMEOUT value has been exceeded for a cached UTOKEN the UTOKEN is deleted from the cache. The next time MQSeries receives a message containing the userid (for which the expired UTOKEN has been deleted from the cache); MQSeries issues the RACROUTE REQUEST=VERIFY call to RACF to create a new UTOKEN for that userid. If RACF returns the new UTOKEN for the userid, MQSeries caches the new UTOKEN in the UTOKEN cache.
  - Caches the UTOKEN returned by RACF, ***without*** the password/PassTicket, in a UTOKEN cache in MQSeries storage.

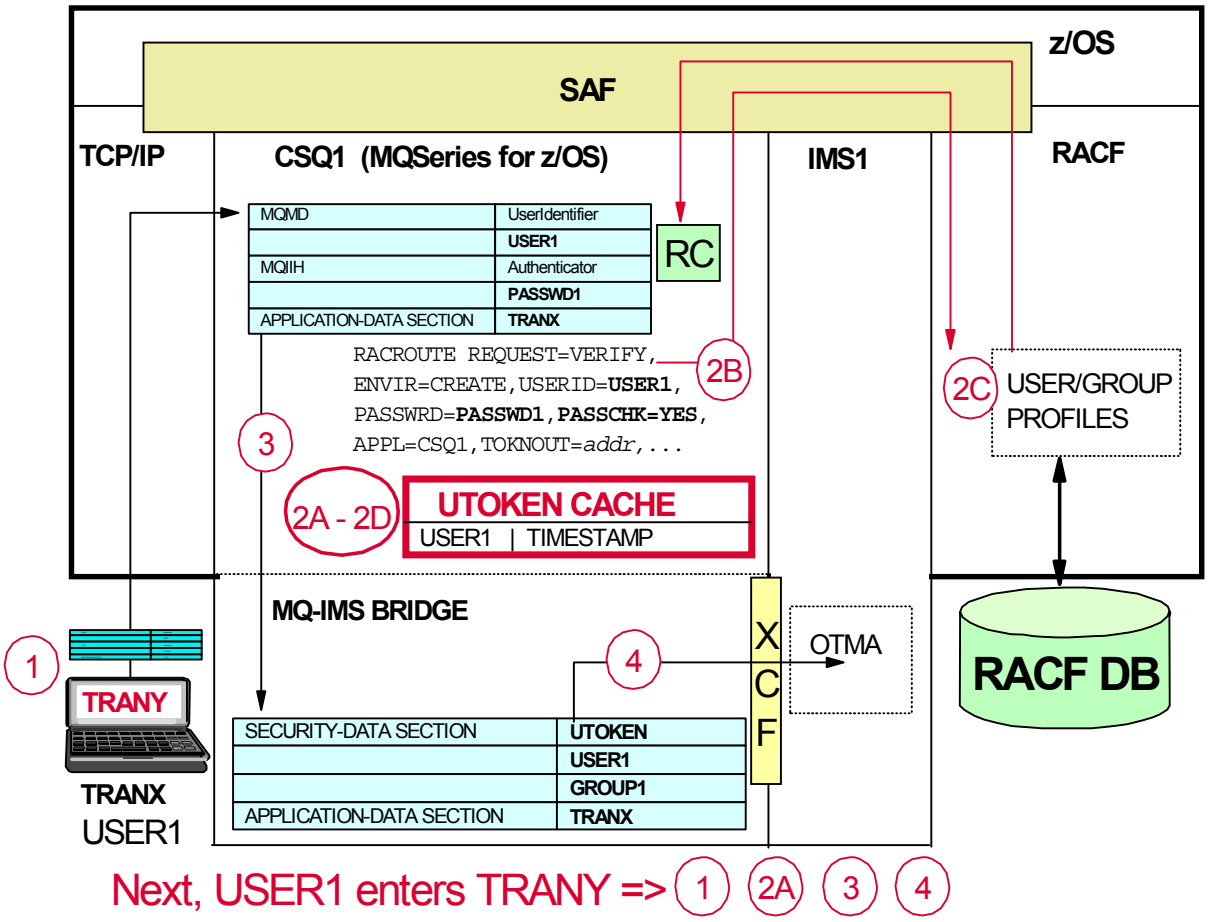
- Use the valid UTOKEN from the cache to verify the userid instead of invoking RACF to perform verification.

  As messages are received from end users, MQSeries checks the UTOKEN cache for a valid UTOKEN prior to invoking RACF to verify the userid and password/PassTicket. If a UTOKEN exists and has not expired, MQSeries avoids a RACROUTE REQUEST=VERIFY call to RACF. The good UTOKEN is placed in the message that is destined for IMS/OTMA via the MQSeries-IMS Bridge application.

MQSeries also offers a type of security called 'REVERIFY' security. Reverify security is activated by the MQSeries **RVERIFY SECURITY** command. The **RVERIFY SECURITY** command is issued to set a reverification flag for all specified userids. Each userid specified on the command is signed off and signed back on again the next time that a request is issued on behalf of that userid that requires a security check. If a request to reverify security has been made, all cached information is lost and a UTOKEN is requested the first time each userid is subsequently encountered.

Figure 15. ACCESS(READ) illustrates the concepts when RACF returns an ACCESS(READ) access level to MQSeries. As you may recall the userid, USER1, had previously entered a message destined for IMS OTMA where TRANX was requested in the message. MQSeries encountered the userid USER1 when the TRANX message was submitted. Now at **step 1**, that same

FIGURE 15. ACCESS(READ)



userid, USER1, enters another message - this time requesting the IMS TRANY transaction code. When the ACCESS(READ) level has been returned to MQSeries, upon receipt of the message MQSeries notes the userid value, USER1, in the UserIdentifier field of the MQMD.

At **step 2A,** MQSeries checks the UTOKEN cache for a UTOKEN for the userid USER1. If MQSeries finds a UTOKEN for USER1 in the cache, the UTOKEN is placed in the message destined for IMS/OTMA, provided that:

- The UTOKEN has not been in the cache longer than the value specified by the TIMEOUT value on the ALTER SECURITY command.
- The userid USER1 was not specified on a RVERIFY SECURITY command.

When MQSeries locates a valid UTOKEN in the cache for the userid (USER1), **step 2B, step 2C, and step 2D** (making an entry in UTOKEN cache for the userid USER1) **are eliminated.**

If MQSeries does not locate a valid UTOKEN in the cache for the userid (USER1), the following is done:

- At **step 2B** a RACROUTE REQUEST=VERIFY macro is issued to RACF. The macro requests that RACF validate the userid USER1, verify the password/PassTicket for USER1, and if both are valid - return a UTOKEN for userid USER1.

- At **step 2C**, RACF verifies the userid USER1 and the corresponding password/PassTicket for USER1. If the combination is valid, RACF returns a UTOKEN to MQSeries for the userid USER1.

- Upon receipt of a UTOKEN from RACF, at **step 2D** MQSeries caches the UTOKEN, ***without*** the password/PassTicket, in the UTOKEN cache. Then the next time USER1 sends a message to MQSeries that is destined for IMS/OTMA, MQSeries may be able to avoid steps 2B through 2D.

For a verified userid, at **step 3** MQSeries places the UTOKEN in the message and passes the message to the MQSeries-IMS Bridge application. The Bridge performs message translation/formating and at **step 4**, the Bridge transmits the message to IMS/OTMA.

ID(*MQ_userid*) ACCESS(UPDATE)

When RACF returns the ACCESS(UPDATE) access level to MQSeries, the following actions are taken:

- A check is made that the userid in the UserIdentifier field of the MQMD structure is known to RACF. The password or PassTicket is ***not*** verified for the userid in the message.

  This means that the RACROUTE REQUEST=VERIFY macro issued by MQSeries specifies PASSCHK=NO and the PASSWRD= keyword is not specified.

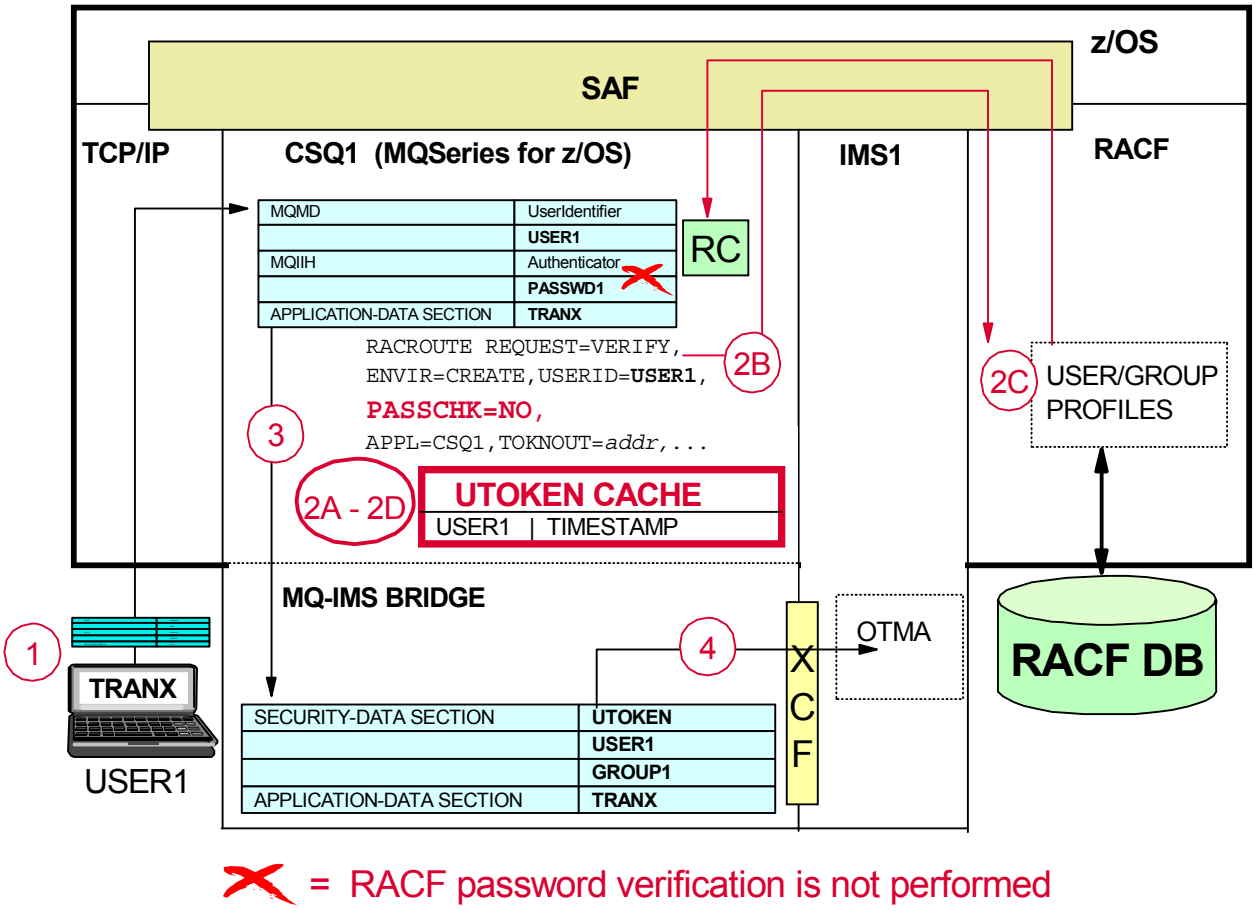
- A UTOKEN is built by RACF and returned to MQSeries. MQSeries does both of the following:

  - Caches the UTOKEN in the MQSeries UTOKEN cache without a password/PassTicket.

  - Passes the UTOKEN to IMS in the message.

Figure 16. ACCESS(UPDATE) illustrates userid verification checking when the ACCESS(UPDATE) level is returned to MQSeries.

FIGURE 16. ACCESS(UPDATE)

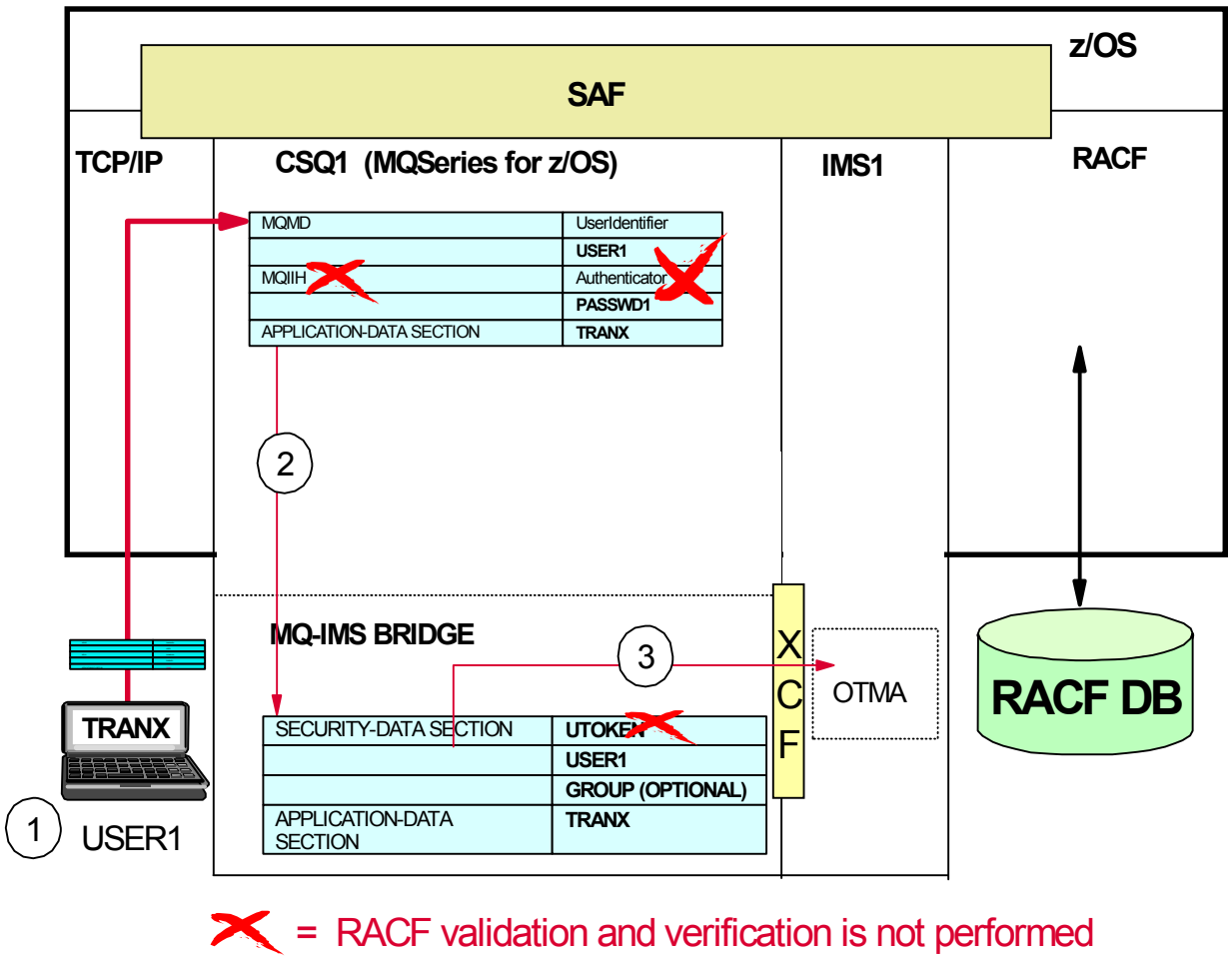


When RACF return the ACCESS(UPDATE) level to MQSeries, the userid verification is similar to that performed for the ACCESS(READ) level. The difference is that when the ACCESS(UPDATE) level is returned, the first time MQSeries encounters a userid (e.g. USER1), the RACROUTE REQUEST=VERIFY macro does not specify a password/PassTicket and the PASSCHK=NO is specified. The PASSCHK=NO specification on the macro causes RACF to validate only the userid, the password (or PassTicket) for the userid is not verified. Otherwise, the verification security checking is the same as for ACCESS(READ).

ID(*MQ_userid*) ACCESS(CONTROL) or ACCESS(ALTER)

Access levels of CONTROL and ALTER indicate that no security UTOKENs need to be provided for any userid in messages destined for an IMS/OTMA system. The userids in messages destined for IMS/OTMA are considered to be 'trusted'. You would probably only use ACCESS(CONTROL) and ACCESS(ALTER) for development and test systems. *Figure 17, 'ACCESS(CONTROL) or ACCESS(ALTER)'* illustrates this point.

FIGURE 17. ACCESS(CONTROL) or ACCESS(ALTER)



When RACF returns an access level of CONTROL or ALTER to MQSeries, userid verification is not performed. As Figure 17 illustrates, at **step 1**, userid USER1 submits a message destined for IMS/OTMA where TRANX is requested.
Upon receipt of the message, **step 2**, MQSeries passes the message to the MQSeries-IMS Bridge application. Note that a UTOKEN cache is not used and a RACROUTE REQUEST=VERIFY call is not made to RACF. The message destined for OTMA will contain security information only of the userid and optionally, the RACF group name. **(NOTE: Although the discussion here is focused on end user messages, the same rules apply to client-bid messages. IMS/OTMA requires OTMA client-bid messages to contain a UTOKEN, unless the OTMA security level is NONE. Therefore, use of CONTROL or ALTER results in the client-bid message being built without a UTOKEN. This will result in a client-bid authorization failure for the MQSeries-IMS Bridge if the OTMA security level is CHECK, FULL, or PROFILE with the 1-byte security flag set to 'C' or 'F'.**

At **step 3**, the Bridge application performs message translation and transmits the message to IMS/OTMA.

Considerations For the Access Level of the MQSeries Userid

The access level that you provide the MQSeries userid on the IMSXCF.XCF_group_name.XCF_member_name_for_IMS FACILITY Class profile:

- Last for the duration of the MQSeries-IMS Bridge application connection. To change the access level requires: updating the FACILITY class profile access list; refreshing the FACILITY class profiles; **stopping OTMA**; and restarting OTMA. You may stop OTMA by issuing the /STOP OTMA command on the IMS system. Conversely, to restart OTMA issue the /START OTMA command on the IMS system. It should be noted that stopping OTMA

may be disruptive to other OTMA client, such as IMS Connect; and will cause the hash tables for all OTMA client to be deleted and rebuilt upon receipt of the client-bid messages.

- PassTickets may be used in lieu of passwords, however the MQSeries-IMS Bridge application does not encrypt nor decrypt the PassTicket.
- Cached UTOKENs are held for the duration defined by the MQSeries ALTER SECURITY command.
- UTOKENs may not be built nor cached by MQSeries if either of the following profiles has been defined to RACF:

qmgr.NO.SUBSYS.SECURITY in the MQADMIN Class. If this profile exist, it could cause the Bridge to fail IMS/OTMA client-bid security if the IMS/OTMA security level is **not** NONE.

IMSXCF.XCF_group_name.XCF_member_name_for_IMS in the FACILITY Class where the MQSeries userid is permitted with ACCESS(CONTROL) or ACCESS(ALTER). If the MQSeries userid has CONTROL or ALTER access: a.) the Bridge's client-bid request will fail if the IMS/OTMA security level is ***not*** NONE; and b.) Security violations may occur in IMS when unverified userids in incoming messages have not been defined to RACF.

Considerations For the MQSeries Client Application Programmer

The content in messages sent from MQSeries client applications to the MQSeries on z/OS is the responsibility of the application programmers that build the messages on the client platform. The messages received from end users by MQSeries on z/OS have similar content as the client-bid message. Messages from end users also contain a SecurityScope field, as shown in the diagram below.

MQMD	
UserIdentifier	USER1
...	
MQIIH	
Authenticator	PASSWD1
SecurityScope	**X**
...	
APPLICATION-DATA SECTION	TRANX
...	

A **1-BYTE** FIELD WHERE *'X'* IS ONE OF THE FOLLOWING VALUES:

- **C**
  - For **CHECK**, which indicates that RACF should be invoked (by IMS/OTMA when the message arrives) to build an ACEE in the control region for the userid in the incoming message
  - If the SecurityScope value is not set by the application programmer, the ***default*** value **C** is used.
- **F**
  - For **FULL**, which indicates that RACF should be invoked (by IMS/OTMA when the message arrives) to build an ACEE in the control region ***and*** a second ACEE in the dependent region for the userid in the incoming message

The SecurityScope field is known to IMS/OTMA as the '1-byte security flag' field that is specified in the security data (SD) section in one of the OTMA message prefixes. It should be noted that **IMS/OTMA**:

- Accepts any one of the possible three (3) values for the SecurityScoope/security-flag field in a message:

  N (for **NONE**, which indicates that RACF should not be called to verify userids in incoming OTMA messages).
  C (for CHECK).
  F (for FULL).

  However, MQSeries only allows the latter two values: C or F. This means that when the IMS/OTMA security level is **PROFILE**, RACF will always be invoked for OTMA messages received the MQSeries-IMS Bridge application. The default value is C (for CHECK).

- Will only **honor** the SecurityScope/security-flag value specification when the IMS/OTMA security level is **PROFILE**.

  Otherwise, when the IMS/OTMA security level is **NONE**, **CHECK** or **FULL**; the security-flag specifications in incoming OTMA messages are ignored.

Now that you understand the considerations for selecting an access level for the MQSeries userid on the IMSXCF profile and for the application programmer setting the SecurityScope value; let's examine the remainder of the steps (which are listed below) to follow when implementing MQSeries-IMS Bridge security:

- Configuring MQSeries to join the same XCF group as the IMS/OTMA system with which the MQSeries-IMS Bridge application will communicate.
- Configuring IMS to join the same XCF group as the MQSeries which will pass messages to IMS/OTMA via the MQSeries-IMS Bridge application.
- Defining the MQSeries objects: storage class, Bridge queue used to transmit messages to IMS/OTMA, and the reply queue used to receive reply messages from IMS/OTMA.
- Operating the MQSeries-IMS Bridge.

Configuring MQSeries to Join the Same XCF Group As IMS/OTMA

This step defines the XCF group and member names for your MQSeries system, and other OTMA parameters. MQSeries and IMS/OTMA must belong to the same XCF group. Use the **OTMACON** statement on the **CSQ6SYSP** macro to tailor these parameters in the system parameter load module. The following parameters are specified on the OTMACON keyword:

- **GROUP** - Name of the XCF group the MQSeries-IMS Bridge application joins
- **MEMBER** - The XCF member name for MQSeries
- **DRUEXIT** - Name of the Destination Resolution User Exit Routine
  The Destination Resolution exit routine (DFSYDRU0) is called to determine the final destination for the output. Each client can specify a separate Destination Resolution exit routine.
- **AGE** - ACEE aging value for the IMS/OTMA hash table for the MQSeries-IMS Bridge
- **TPIPEPREF** - First 3 characters of the transaction pipe prefix name
  A transaction pipe (TPIPE) is the logical connection between the IMS/OTMA server and the OTMA client, such as the MQSeries-IMS Bridge. An OTMA client includes the transaction-pipe name in the message-control information section of the message prefix for the input message. IMS then associates application output for an OTMA client with a specific transaction pipe.

MQSeries provides some recommendations for coding the parameter specifications on the OTMACON statement. The recommendations are listed in the table below.

OTMACON(group,member,druexit,age,tpipepref)	
group	**XCF group name.** Required parameter when the MQSeries-IMS Bridge is to be used. Default is blanks for group name. MQSeries-IMS Bridge is not started when group name blanks.
member	**MQSeries XCF member name.** Default is queue manager name (i.e. CSQ1).
druexit	**Destination Resolution User Exit name.** Default name in IMS/OTMA is DFSYDRU0. MQSeries suggests changing exit name to identify queue mannager using DRU exit. Suggested exit naming convention: DRU0**CSQ1**, DRU0**CSQ2**, DRU0**CSQ3**, ...
age	**ACEE aging value.** Number of seconds an ACEE (created for a MQSeries-IMS Bridge end user) is valid in IMS.
tpipepref	**First 3 characters of transaction pipe (TPIPE) prefix name.** Default is CSQ. Do not change default TPIPE prefix name without good reason.

Configuring IMS to Join the Same XCF Group as the MQSeries-IMS Bridge

This step defines the XCF group and member names for the IMS system. Add the following parameters to your IMS parameter list, either in your JCL or in member DFSPBxxx in the IMS.PROCLIB:

- **GRNAME=**

  Specifies the XCF group IMS is to join. The group name is one to eight uppercase alphanumeric characters or other valid characters ($, @). IMS joins the XCF group either during IMS initialization (if OTMA=Y is specified) or as a result of an IMS /START OTMA command. If GRNAME= is not specified and OTMA=N is specified, IMS cannot join the XCF group.

- **OTMA=**

  Specifies that the IMS Open Transaction Manager Access (OTMA) function is to be enabled during IMS initialization. Valid values are Y (yes) or N (no). The default value is N . If Y is specified, IMS attempts to create the OTMA group during initialization and then attempts to join that group. **MQSeries recommends that IMS/OTMA users start IMS using OTMA=N** so that OTMA will not be flooded with messages from MQSeries. After the MQSeries-IMS Bridge has joined the XCF group, issue the /START OTMA command from IMS to start OTMA.

- **OTMANM=**

  Is used to specify the XCF member name that IMS uses for the group when IMS is not using XRF or RSR. The member name is 1 to 16 uppercase alphanumeric characters or other valid characters ($, @). The OTMANM name can be specified in the IMS procedure or in the DFSPBxxx member.

  If OTMANM is not specified, IMS uses the APPLID= value (supplied by the COMM macro used to generate IMS) for the IMS XCF member name. The **APPLID1=** value may be used to override the COMM macro APPLID= value.

  If IMS is using XRF or RSR, the XCF member name that IMS uses comes from the **USERVAR=** name specified in the IMS procedure, in the DFSPBxxx member, or in the DFSHSBxx member. The OTMANM name is not used in this case.

Defining the MQSeries Objects

You tell MQSeries the XCF group and member name of the IMS system by defining the MQ objects. This is specified by the storage class of a queue. If you want to send messages across the MQSeries-IMS bridge you need to specify this when you define the storage class for the queue.

In the storage class, you need to define the XCF group and the member name of the target IMS system. To do this, either use the MQSeries operations and control panels, or use the MQSC commands as described in the 'MQSeries Command Reference' manual.

Sample storage class, Bridge queue, and reply queue definitions are shown below.

Storage class definition

```
DEFINE STGCLASS( 'BRIDGE' ) +
XCFGNAME( 'IMSOTMA' ) +
XCFMNAME ( 'IMS1MEM' ) +
PSID( 02 )
```

'Bridge' queue definition

```
DEFINE QLOCAL(BRIDGE.TEST.IMS.QUEUE) REPLACE -
DESCR  ( 'MQ-IMS bridge queue' ) -
STGCLASS(BRIDGE)
```

Reply queue definition

```
DEFINE QLOCAL(BRIDGE.TEST.IMS.REPLYTO.QUEUE) REPLACE -
DESCR ( 'Queue used for reply messages' ) -
STGCLASS(REPLY)
```

Operating the MQSeries-IMS Bridge

There are no MQSeries commands to control the MQSeries-IMS bridge. Start the IMS bridge by starting OTMA. Either use the IMS command /START OTMA, or start it automatically by specifying OTMA=Y in the IMS system parameters in DFSPBxxx member of IMS.PROCLIB. If OTMA is already started, the bridge starts automatically when MQSeries startup has completed. An MQSeries event message is produced when OTMA is started.

Use the IMS command /STOP OTMA to stop OTMA communication. When this command is issued, an MQSeries event message is produced.

Summary

This document has provided a technical overview of the following

- The IMS/OTMA security options (NONE, CHECK, FULL, and PROFILE). Additionally, client-bid security, command authorization, and transaction authorization have been described for each of the OTMA security levels. Where appropriate, information on user security exit routines, such as the Transaction Authorization Exit, has been presented..

  Customer requirements for ***IMS-TM/OTMA*** environments and an update on the status of security-related enhancements that are planned were presented

- The security capabilities of IMS Connect and a user security exit routine have been presented.

  Customer requirements for ***IMS Connect*** and an update on the status of security-related enhancements that are planned for IMS Connect were presented.

- The MQSeries-IMS Bridge application security concepts when the Bridge is used to access IMS/OTMA.

If you require additional information, please contact one of the authors of this document.