

AS/400 Advanced Series



AnyMail/400 Mail Server Framework Support

Version 4

AS/400 Advanced Series



AnyMail/400 Mail Server Framework Support

Version 4

Take Note!

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

First Edition (August 1997)

This edition applies to the licensed program IBM Operating System/400 (Program 5769-SS1), Version 4 Release 1 Modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions.

Make sure that you are using the proper edition for the level of the product.

Order publications through your IBM representative or the IBM branch serving your locality. If you live in the United States, Puerto Rico, or Guam, you can order publications through the IBM Software Manufacturing Solutions at 800+879-2755. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication. You can also mail your comments to the following address:

IBM Corporation
Attention Department 542
IDCLERK
3605 Highway 52 N
Rochester, MN 55901-7829 USA

or you can fax your comments to:

United States and Canada: 800+937-3430
Other countries: (+1)+507+253-5192

If you have access to Internet, you can send your comments electronically to IDCLERK@RCHVMW2.VNET.IBM.COM; IBMMAIL, to IBMMAIL(USIB56RZ).

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

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

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

Contents

Notices	v	Chapter 5. Mail Server Framework Message Concepts	5-1
Programming Interface Information	v	MSF Message Lists	5-2
Trademarks and Service Marks	v	Originator List	5-2
		Original Recipient List	5-2
		Recipient List	5-2
		Envelope List	5-3
		Attachment Reference List	5-3
		Other Lists	5-3
		Report-On List	5-3
		Report-To List	5-3
		Message Identifier	5-3
		MSF Data Types	5-4
		Rules for MSF Data Type Definitions	5-4
		How the MSF Data Types Are Used to Call Exit Point Programs	5-4
		How the MSF Message Changes	5-4
		How the MSF Message Flows Through the Framework	5-5
Chapter 1. AnyMail/400 Mail Server Framework - Introduction	1-1	Appendix A. Mail Server Framework APIs	A-1
		MSF Configuration APIs	A-1
		Add Mail Server Framework Configuration (QzmfAddMailCfg) API	A-1
		List Mail Server Framework Configuration (QzmfLstMailCfg) API	A-1
		Remove Mail Server Framework Configuration (QzmfRmvMailCfg) API	A-1
		MSF Message APIs	A-1
		Create Mail Message (QzmfCrtMailMsg) API	A-1
		Change Mail Message (QzmfChgMailMsg) API	A-2
		Retrieve Mail Message (QzmfRtvMailMsg) API	A-2
		Other Optional APIs	A-2
		Query Mail Message Identifier (QzmfQryMailMsgId) API	A-2
		Reserve Mail Message Identifier (QzmfRsvMailMsgId) API	A-2
		Complete Creation Sequence (QzmfCrtCmpMailMsg) API	A-2
		Remove Reserved Mail Message Identifier (QzmfRmvRsvMailMsgId) API	A-2
		Exit Points	A-2
		Snap-In Call Exit Point Program	A-2
		Track Mail Message Changes Exit Point Program	A-2
		Validate Data Field Exit Point Program	A-2
Chapter 2. Operations Considerations	2-1	Appendix B. MSF Journal Logging and Journal Formats	B-1
Starting and Stopping the Mail Server Framework	2-1	QZMF Journal	B-1
The Start Mail Server Framework (STRMSF) Command	2-1	Displaying the QZMF Journal	B-1
The End Mail Server Framework (ENDMSF) Command	2-1	Displaying the Journal Entries	B-2
MSF QSYSWRK Jobs	2-2	Entry Type LG	B-3
Managing MSF Jobs	2-2	Entry Type ER	B-3
Description Considerations	2-3	Entry Type SY	B-3
Descriptions Reset During OS/400 Installation	2-3	Entry Type CF	B-4
Changing Subsystem and Job Descriptions	2-3	Format for MSF Message Logging (LG)	B-5
Error Recovery and Error Messages	2-3	Format for MSF Message Errors (ER)	B-7
QMSF Job Error Recovery	2-3		
STRMSF and ENDMSF Command Error Recovery	2-4		
Chapter 3. Configuration Considerations	3-1		
Exit Point Program Registration	3-1		
MSF Configuration APIs	3-2		
Chapter 4. Mail Server Framework Structure	4-1		
Exit Points	4-1		
Exit Point Programs	4-1		
Snap-Ins	4-1		
Exit Points Provided on the Mail Server Framework	4-2		
MSF Exit Point Groups	4-3		
Exit Point Programs in the Addressing Group	4-4		
List Expansion Exit Point	4-4		
Address Resolution Exit Point	4-4		
Exit Point Programs in the Pre-Delivery Processing Group	4-6		
Envelope Processing Exit Point	4-6		
Attachment Conversion Exit Point	4-6		
Security and Authority Exit Point	4-7		
Exit Point Programs in the Delivery Group	4-8		
Local Delivery Exit Point	4-8		
Message Forwarding Exit Point	4-8		
Exit Point Programs in the Management Group	4-9		
Non-Delivery Exit Point	4-9		
Attachment Management Exit Point	4-9		
Accounting Exit Point	4-9		

Format for MSF System Level Events Table (SY)	B-8
Format for MSF Configuration Changes (CF)	B-9
Appendix C. Preregistered Exit Point Programs . . .	C-1
Exit Program QZMFSNPA	C-1
Appendix D. How Mail Server Framework Works with SNADS, Object Distribution, and OfficeVision	D-1
List Expansion	D-3
Address Resolution	D-3

Local Delivery	D-3
Message Forwarding	D-3
Non-Delivery	D-3
Attachment Management	D-4
Accounting	D-4

Bibliography	H-1
-------------------------------	-----

Index	X-1
------------------------	-----

Figures

1-1. MSF Support when Sending Electronic Mail	1-3	5-2. MSF Message Lists	5-2
1-2. MSF Support when Receiving Electronic Mail . . .	1-4	5-3. MSF Data Types Used to Determine Which Exit Point Program to Call	5-5
2-1. QMSF Jobs in QSYSWRK Subsystem	2-2	5-4. Exit Point Programs Can Change Information in an MSF Message	5-5
2-2. Work with Subsystems Display	2-3	5-5. How an MSF Message Flows Through the Framework	5-6
2-3. Work with Subsystem Jobs Display	2-3	B-1. Display Journal - First Display	B-1
3-1. Work with Registration Information - First Display	3-1	B-2. Display Journal - Second Display	B-1
3-2. Work with Registration Information - Second Display	3-1	B-3. Display Journal - Third Display	B-2
3-3. Work with Registration Information Display	3-1	B-4. Display Journal Entries Display	B-2
3-4. Work with Exit Programs Display	3-2	B-5. Display Journal Entry - Type LG	B-3
3-5. Display Exit Program Display	3-2	B-6. Display Journal Entry - Type ER	B-3
3-6. Example of an Exit Program in the Mail Server Framework	3-2	B-7. Display Journal Entry - Type SY	B-3
4-1. MSF Exit Point and Exit Point Program Overview	4-2	B-8. Display Journal Entry - Type CF	B-4
4-2. Using Multiple MSF Exit Point Programs	4-2	B-9. Fields of the Journal Entries - Example	B-6
4-3. MSF Exit Point Groups	4-3	B-10. Function Identifier A - Example	B-11
4-4. Exit Point Programs in the Addressing Group . . .	4-4	B-11. Function Identifier B - Example	B-11
4-5. Exit Point Programs in the Pre-Delivery Processing Group	4-6	B-12. Function Identifier D - Example	B-11
4-6. Exit Point Programs in the Delivery Group	4-8	B-13. Function Identifier F - Example	B-11
4-7. Exit Point Program in the Management Group . . .	4-9	C-1. Exit Point Programs Shipped with MSF	C-1
5-1. MSF Message Concept Overview	5-1	C-2. Address Resolution Exit Point Programs	C-1
		C-3. Example of Preferred Address Resolution . . .	C-2
		D-1. SNADS Overview	D-2

Tables

4-1. MSF Exit Points	4-1	B-3. Type SY Journal Entries for the MSF System Level Events Table Changes	B-8
B-1. Type LG Journal Entries for MSF Messages	B-5	B-4. Type CF Journal Entries for MSF Configuration Changes	B-9
B-2. Type ER Journal Entries for MSF Message Errors	B-7		

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact the software interoperability coordinator. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Address your questions to:

IBM Corporation
Software Interoperability Coordinator
3605 Highway 52 N
Rochester, MN 55901-7829 USA

This publication could contain technical inaccuracies or typographical errors.

This publication may refer to products that are announced but not currently available in your country. This publication may also refer to products that have not been announced in your country. IBM makes no commitment to make available any unannounced products referred to herein. The final decision to announce any product is based on IBM's business and technical judgment.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This publication contains small programs that are furnished by IBM as simple examples to provide an illustration. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. All programs contained herein are provided to you "AS IS". THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.

Programming Interface Information

This book is intended to help the system operator and system administrator use the mail server framework on the IBM AS/400 system. It also provides concepts for the business partner or system programmer. This book documents General-Use Programming Interface and Associated Guidance Information provided by the Operating System/400 licensed program.

General-Use Programming Interfaces allow the customer to write programs that obtain the services of the mail server framework.

Trademarks and Service Marks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Application System/400	Operating System/400
AS/400	OS/400
IBM	400
OfficeVision	

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

About AnyMail/400 Mail Server Framework Support (SC41-5411)

This book contains information about the mail server framework (MSF) on the AS/400 system. The book is organized as follows:

- Introduction to the mail server framework (Chapter 1, "AnyMail/400 Mail Server Framework - Introduction").
- Usage considerations for the mail server framework (Chapter 2, "Operations Considerations"). The control language (CL) commands used to start and end the framework operation are in this chapter. There is also information about error messages and error recovery.
- Configuration considerations for the mail server framework (Chapter 3, "Configuration Considerations"). This chapter explains how to find the registered exit points and how to register exit programs with an exit point.
- Description of the mail server framework structure (Chapter 4, "Mail Server Framework Structure"). This includes definitions of exit points and exit point programs.
- Conceptual information about the mail server framework messages (Chapter 5, "Mail Server Framework Message Concepts"), including MSF message lists and MSF data types.
- Overview of mail server framework APIs (Appendix A, "Mail Server Framework APIs").
- Information about the QZMF journal, journal formats, and journal logging (Appendix B, "MSF Journal Logging and Journal Formats").
- Description of the preregistered exit point programs that are shipped with MSF (Appendix C, "Preregistered Exit Point Programs").
- Description of how SNA distribution services (SNADS) uses the mail server framework to route and distribute messages (Appendix D, "How Mail Server Framework Works with SNADS, Object Distribution, and OfficeVision").

The term *operator* refers to the system operator as the person responsible for operating the console, responding to messages, and diagnosing any system malfunctions (problem analysis).

For a list of publications related to this book, see the "Bibliography."

Who Should Use This Book

This book supplies the system operator and system administrator with the information needed to use the mail server framework on the AS/400 system. The operator's role is to maintain and support mail on the AS/400 system. The operator may find the following parts of this book most useful when using the framework:

Chapter 1, AnyMail/400 Mail Server Framework - Introduction
Chapter 2, Operations Considerations
Chapter 3, Configuration Considerations
Appendix B, MSF Journal Logging and Journal Formats
Appendix C, Preregistered Exit Point Programs
Appendix D, How Mail Server Framework Works with SNADS, Object Distribution, and OfficeVision

This book also supplies the business partner or system programmer with conceptual information about the mail server framework. Use the following parts of this book to better understand how the mail server framework is used:

Chapter 1, AnyMail/400 Mail Server Framework - Introduction
Chapter 3, Configuration Considerations
Chapter 4, Mail Server Framework Structure
Chapter 5, Mail Server Framework Message Concepts
Appendix A, Mail Server Framework APIs

Benefits of Using the Mail Server Framework Support

The mail server framework provides the basis for OfficeVision mail, SNA distribution services (SNADS), and any future mail offering on the AS/400 system. The QMSF mail server framework jobs, running in the QSYSWRK subsystem, support the mail server framework.

Users of the mail server framework can tailor their mail applications to meet their specific needs. The user applications are enabled and managed by the mail server framework.

Prerequisite and Related Information

For information about other AS/400 publications (except Advanced 36), see either of the following:

- The *Publications Reference* book, SC41-5003, in the AS/400 Softcopy Library.
- The *AS/400 Information Directory*, a unique, multimedia interface to a searchable database that contains descriptions of titles available from IBM or from selected other publishers. The *AS/400 Information Directory* is shipped with the OS/400 operating system at no charge.

Information Available on the World Wide Web

More AS/400 information is available on the World Wide Web. You can access this information from the AS/400 home page, which is at the following uniform resource locator (URL) address:

<http://www.as400.ibm.com>

Select the Information Desk, and you will be able to access a variety of AS/400 information topics from that page.

Chapter 1. AnyMail/400 Mail Server Framework - Introduction

The AnyMail/400 mail server framework is an open structure for electronic mail distribution that is provided with the OS/400 operating system. AnyMail/400 functions are a set of mail-related functions that provide open and flexible interfaces to support mail on the AS/400. The mail server framework (MSF) is the distribution framework. The primary function of the mail server framework is to allow the distribution of electronic mail messages (for example, voice, video, or text).

On the AS/400 system, the mail server framework performs the following functions:

- Creates and queues MSF messages
- Distributes MSF message information by calling configured exit programs

An **exit point** is a specific point in a system function or program where control is passed to one or more specified programs. An exit point passes control to an **exit program**. The exit program can be written by the user or it can be a program that is already on the system. An **application program interface (API)** is a functional interface supplied by the operating system that allows an application program written in a high-level language to use specific data or functions of the operating system.

The mail server framework allows programs to be configured and accessed through an API. A **snap-in** is a registered exit point program that is called from a mail server framework exit point. The user-written exit programs provide the mail message processing required for each MSF message that is created.

The mail server framework (MSF) message contains information that defines the electronic mail message. The mail server framework does not process the contents of a mail message. Instead, it determines which exit program is allowed to work with the MSF message and it tracks the flow of the MSF message through the framework. When an MSF message is sent, the MSF data type is stored and used asynchronously. The framework also provides support for electronic mail messages that arrive from other systems.

Figure 1-1 on page 1-3 illustrates client mail enabled applications using the AS/400 system as a server and an AS/400 application, such as OfficeVision. These applications are creating MSF messages and the exit point programs configured in the framework use the information in the messages to send the information to applications on other systems.

Figure 1-2 on page 1-4 illustrates electronic mail information arriving from applications on other systems. The support for the information arriving from specific protocols uses the MSF by creating MSF messages. The framework then uses exit point programs to deliver the message information to a

system supported message store. The client and the AS/400 application would then read the information from the message store.

The MSF message might be created by an application and, as a result of using specific exit point programs, the information might be sent to other applications on the same system, as well as applications on other systems. Also, messages could be arriving from other systems. This is not shown in Figure 1-1 on page 1-3 or Figure 1-2 on page 1-4. As a result of using specific exit point programs, that information could be both delivered to the application on that system and forwarded to another system.

The exit points allow you to install software to affect MSF message flow through the mail server framework. For each of the mail server framework exit points provided, functions of the exit programs follow:

List expansion

Allows for expansion of the distribution lists for the MSF message. It expands any distribution list found in a recipient list. Note that distribution lists may contain other distribution lists that can be expanded further.

Address resolution

Allows for address mapping for each destination address and adds the correct information to the MSF message recipient information so appropriate exit programs can be called at later exit points.

Envelope processing

Allows the message envelope to be changed to a format that the mail delivery system accepts.

Attachment conversion

Allows for changes or additions to MSF message attachments.

Security and authority

Allows for verification of user authority to distribute the MSF message information to addresses in its recipient list, before the MSF message is delivered using the local or remote delivery exit programs.

Local delivery

Allows for delivery of the MSF message to local recipients.

Message forwarding

Allows forwarding of the MSF message information to remote recipients.

Non-delivery

Allows for reporting of the non-delivery of an MSF message.

Attachment management

Allows for management of attachments referenced by an MSF message as part of the message distribution.

Accounting

Allows for an audit trail of activity required in the electronic mail environment.

See Chapter 4, "Mail Server Framework Structure" for more information about each of the exit points.

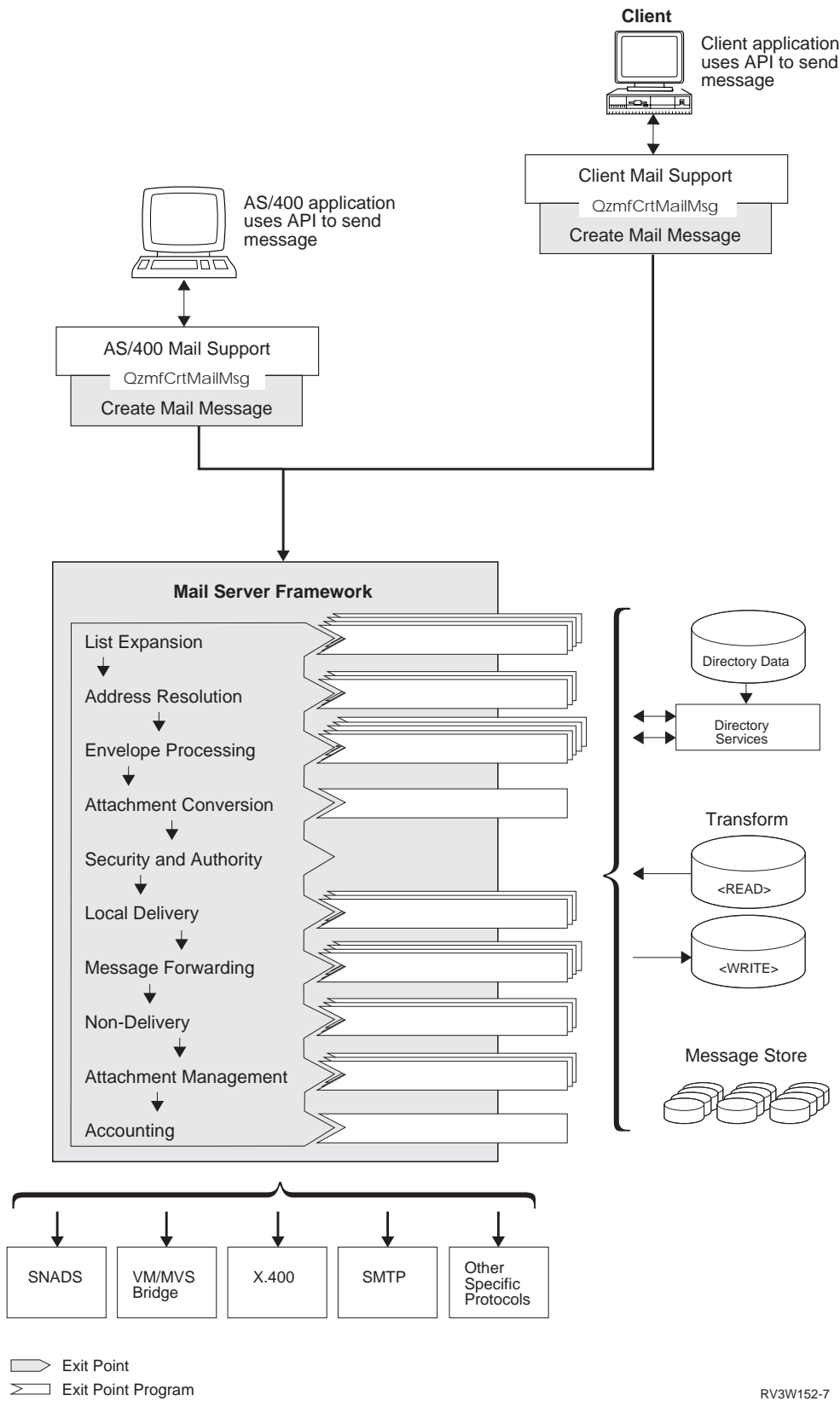


Figure 1-1. MSF Support when Sending Electronic Mail

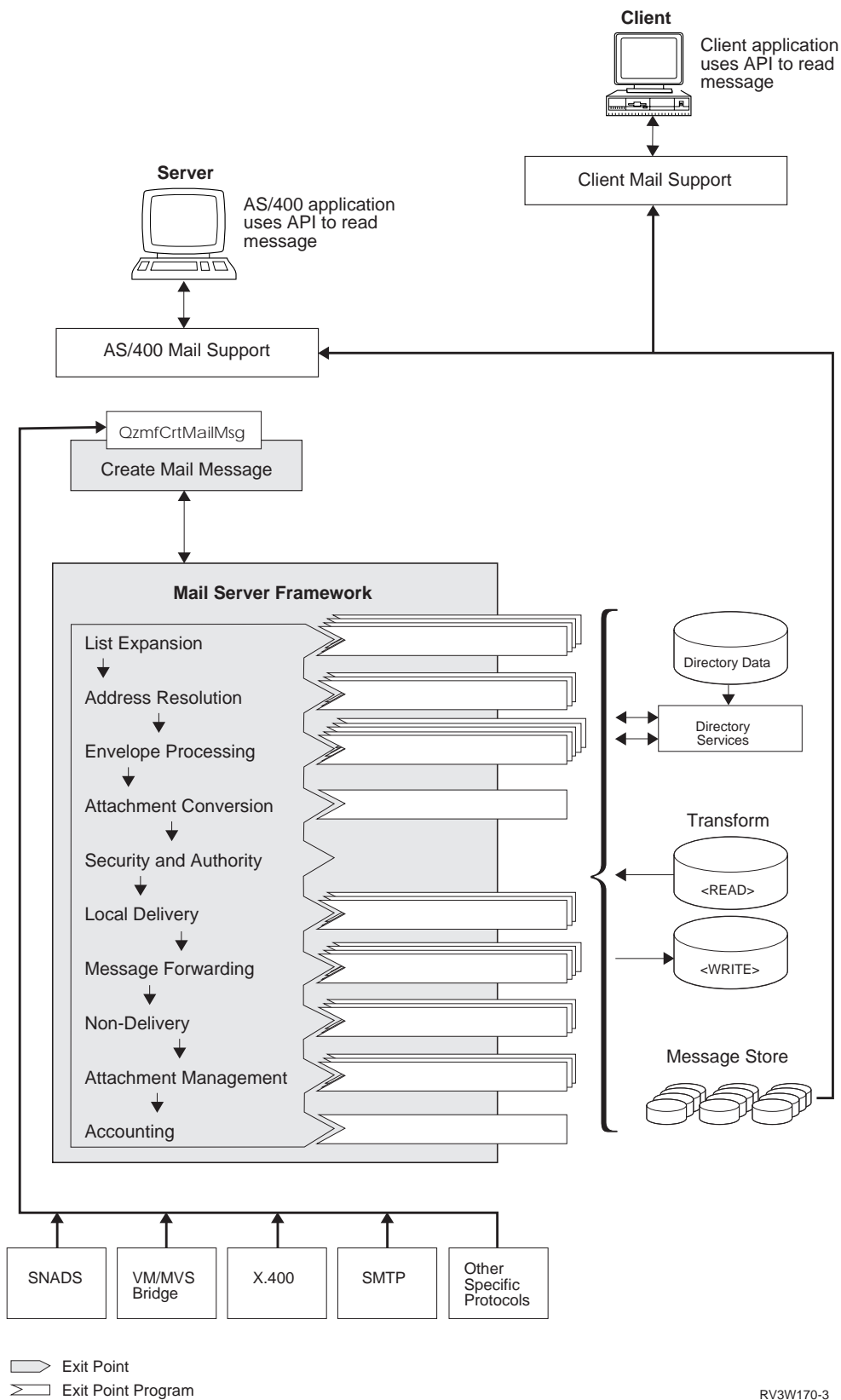


Figure 1-2. MSF Support when Receiving Electronic Mail

Chapter 2. Operations Considerations

If you use applications such as OfficeVision/400 or object distribution, make sure that both the SNADS subsystem and the mail server framework are active.

Starting and Stopping the Mail Server Framework

There are two control language (CL) commands that control the processing of MSF messages:

- Start Mail Server Framework (STRMSF)
- End Mail Server Framework (ENDMSF)

The primary purposes of the STRMSF and ENDMSF commands are to start and end the framework, and to reset the framework when errors occur.

The Start Mail Server Framework (STRMSF) Command

The Start Mail Server Framework (STRMSF) command starts the mail server framework jobs in the system work subsystem (QSYSWRK). For more information about QSYSWRK, see the *Work Management* book. There are two parameters you can specify on the STRMSF command.

MSGOPT (Message Option)

This parameter specifies how the mail server framework processes existing MSF messages.

*RESUME is the default option, which allows all existing MSF messages to continue processing from the point the mail server framework was previously ended.

*RESET allows all existing MSF messages to be processed as if they were just created. Users may receive duplicate messages when this parameter is used.

Note

*CLEAR deletes all existing MSF messages. Use the *CLEAR option only when a software error is reported with the mail server framework or its associated exit point programs that requires all MSF messages to be deleted. Use the option only when you want to remove all mail message traffic from the system because of errors. The possibility may exist that a message was already delivered before an error occurred. If this option is used, all messages are deleted and cannot be recovered.

NBRMSFJOB (Number of Mail Server Framework Jobs)

This parameter specifies the number of mail server framework jobs to start in the QSYSWRK subsystem. Several MSF messages can be processed concurrently.

The valid value range is 1 through 99. If mail traffic is high, use a larger value. Performance may be affected by the amount of mail traffic occurring.

The default for this parameter is 3.

Example 1: Starting One Mail Server Framework Job

```
STRMSF NBRMSFJOB(1)
```

This example starts one QMSF job (in the QSYSWRK subsystem) in a normal manner, processing any MSF messages at the point at which processing was interrupted.

Example 2: Restarting Mail Server Framework Jobs

```
STRMSF MSGOPT(*RESET)
```

This example restarts the MSF jobs, using the default of 3. The MSF messages, partially handled by previous MSF jobs, are processed again as if they were just created.

The End Mail Server Framework (ENDMSF) Command

The End Mail Server Framework (ENDMSF) command ends the mail server framework jobs in the system work subsystem (QSYSWRK). This command stops all MSF message distribution; however, MSF messages can continue to be created even when the QMSF jobs are intentionally ended or have failed. These messages will be processed when the QMSF jobs are restarted by the STRMSF command.

There are two parameters you can specify on the ENDMSF command.

OPTION (Option)

This parameter specifies whether the mail server framework jobs in QSYSWRK end immediately or in a controlled manner.

*CNTRLD allows all MSF jobs to end in a controlled manner. This option allows each framework job a chance to process the MSF messages completely before the job ends. *CNTRLD is the default.

*IMMED allows all MSF jobs to end immediately. The MSF messages being processed when the jobs end are reprocessed when the mail server framework is restarted.

DELAY (Delay)

This parameter can be used to specify the maximum amount of delay time, in seconds, before the MSF jobs are ended.

This parameter is ignored if OPTION(*IMMED) is specified.

Valid values range from 1 through 999999 seconds.

The default option for this parameter is 30 seconds, which means that a maximum delay time of 30 seconds is allowed before the MSF jobs are ended.

Example 1: Ending the Mail Server Framework in a Controlled Manner

```
ENDMSF OPTION(*CNTRLD) DELAY(60)
```

This example ends the MSF jobs in the QSYSWRK subsystem in a controlled manner. The MSF jobs have 60 seconds to complete processing any MSF messages currently being handled.

Example 2: Ending the Mail Server Framework Immediately

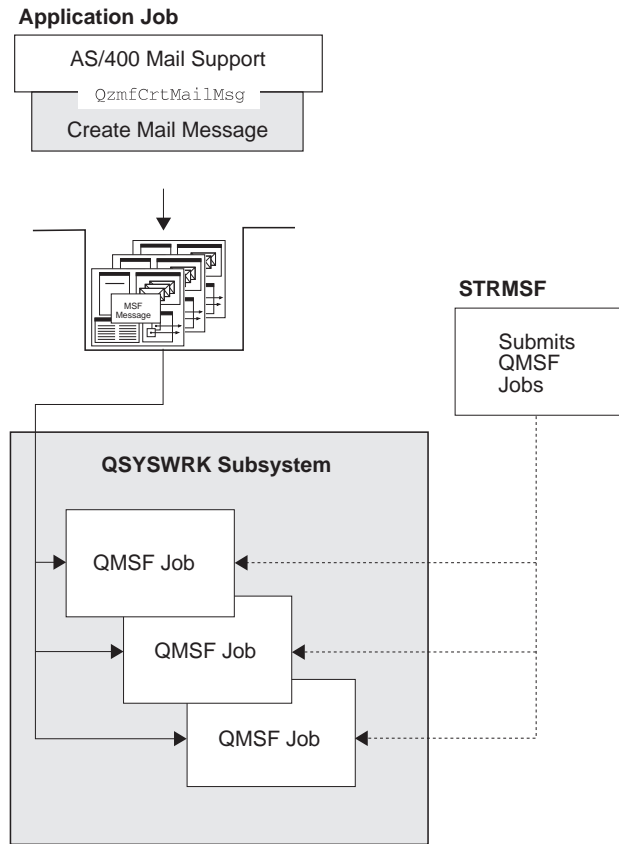
```
ENDMSF OPTION(*IMMED)
```

This example ends the MSF jobs in the QSYSWRK subsystem immediately. The MSF jobs do not complete processing of any MSF messages currently being handled.

MSF QSYSWRK Jobs

The STRMSF command starts a specified number of jobs in the QSYSWRK subsystem named QMSF.

These jobs perform the distribution function of the mail server framework. Note MSF messages can be created in other jobs.



RV3W172-2

Figure 2-1. QMSF Jobs in QSYSWRK Subsystem

Managing MSF Jobs

There is an autostart entry, QZMFECOX, that is shipped as part of the QSYSWRK subsystem. Whenever the QSYSWRK subsystem is started, one QMSF job is started by the autostart job entry.

If you want to change the number of jobs that are started in the QSYSWRK subsystem, change the STRMSF command that is in the request data of the QSYS/QZMFEJBD job description. Use the Change Job Description (CHGJOB) command to make the change.

If you do not want any MSF jobs started when the QSYSWRK subsystem starts, remove the autostart job entry, QZMFECOX, from the QSYSWRK subsystem description. Use the Remove Autostart Job Entry (RMVAJE) command.

For more information about QSYSWRK and work management, refer to the *Work Management* book.

To work with the QMSF jobs, first locate the QSYSWRK subsystem where they are running. Use the Work with Subsystem display. Type WRKSBS on the command line and press the Enter key. Then type 8 in the *Opt* field next to QSYSWRK.


```

Work with Subsystems
System: RCHASEEL

Type options, press Enter.
4=End subsystem 5=Display subsystem description
8=Work with subsystem jobs

-----Subsystem Pools-----
Opt Subsystem Total Storage (K) 1 2 3 4 5 6 7 8 9 10
--- BATS 0 2
--- QBATCH 0 2
--- QCMN 0 2
--- QCTL 0 2
--- QINTER 0 2 4
--- QSNADS 0 2
--- QSPL 0 2 3
.8_ QSYSWRK 0 2
--- QXFPCS 0 2

Parameters or command
====>
F3=Exit F5=Refresh F11=Display system data F12=Cancel
F14=Work with system status
Bottom

```

Figure 2-2. Work with Subsystems Display

QSYSWRK contains several jobs relating to the OS/400 operating system. You can expect to see at least one QMSF job running in the QSYSWRK subsystem when it is started.

The reason for having more than one QMSF job is throughput performance. If you have only a very few mail users on your system, consider reducing the number of QMSF jobs. However, if the mail processing appears to be very slow, consider starting more QMSF jobs.

Figure 2-3 shows QSYSWRK with QMSF and other jobs that typically run there. In this particular example, there are three QMSF jobs running.

The MSF jobs are named QMSF. To work with a QMSF job, use option 5.

```

Work with Subsystem Jobs
System: RCHASEEL 01/16/95 15:53:21
Subsystem . . . . . : QSYSWRK

Type options, press Enter.
2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
8=Work with spooled files 13=Disconnect

Opt Job User Type -----Status----- Function
--- QDIRSHDCTL QDOC BATCH ACTIVE * -DIRSHD
5_ QMSF QMSF BATCH ACTIVE
--- QMSF QMSF BATCH ACTIVE
--- QMSF QMSF BATCH ACTIVE
--- QTCPIP QSYS BATCH ACTIVE
--- QTFTP01045 QTCP BATCH ACTIVE CMD-RGZPFM
--- QTFTP01231 QTCP BATCH ACTIVE CMD-MOVOBJ
--- QTFTP01956 QTCP BATCH ACTIVE CMD-MOVOBJ
--- QTFTP10209 QTCP BATCH ACTIVE CMD-MOVOBJ
--- QTFTP27001 QTCP BATCH ACTIVE CMD-RGZPFM

Parameters or command
====>
F3=Exit F4=Prompt F5=Refresh F9=Retrieve F11=Display schedule data
F12=Cancel
More...

```

Figure 2-3. Work with Subsystem Jobs Display

Description Considerations

Following are tips to consider when you are working with subsystem descriptions and job descriptions.

Descriptions Reset During OS/400 Installation

When the OS/400 licensed program is installed, all subsystem descriptions and job descriptions get replaced. Therefore, if either QSYSWRK subsystem description or QZMFEJBD job description is changed before an installation of OS/400, the changes will be lost after the installation is completed. If changes to the descriptions need to be saved, the job description and subsystem description objects should be saved before the installation. These objects can then be restored after the installation completes. Use the Save Object (SAVOBJ) and Restore Object (RSTOBJ) commands.

Changing Subsystem and Job Descriptions

If you change the subsystem description and job description, the subsystem may not start or any jobs associated with MSF may fail to run.

No log entries are made to indicate there is an error.

Error Recovery and Error Messages

QMSF Job Error Recovery

If MSF messages are not processed: Situations may occur in which you find that messages are not being processed. Error messages or job log messages may indicate that the mail server framework stopped the processing of some messages and is beginning to process new messages from the message queue.

To recover from this situation, check the job logs for the exit point and the exit program running when MSF stopped processing the messages. Use the information to determine where and why MSF stopped processing the messages.

If the mail server framework ends: If the MSF job in QSYSWRK ends, first try restarting MSF. If that does not resolve the problem, check for messages in QSYSOPR and in the job logs for the QMSF jobs that are running that caused MSF to end. The mail server framework is finding something it cannot process, so it ends the framework.

A case where this could occur is if someone deleted the exit point program, but did not remove it (unregister the program) from the exit point. You can use the Work with Registration Information (WRKREGINF) display to check for this situation.

If these messages appear in the job log:

- CPFAF95 - Job ended
 - Exit programs encountered severe errors causing the job to end.
 - The mail server framework encountered abnormal conditions.
 - Exit programs encountered situations that stopped the processing of MSF messages and ended the MSF job.

- CPFAF98 - Message postponed

One of the exit programs determined that an MSF message should be postponed until the next STRMSF.

To recover from these messages, use the Display Job Log (DSPJOBLOG) command to determine what the problem is. Display the message and use the recovery information to correct the problem. Then restart MSF.

See the *Alerts Support* book for a list of the IBM-supplied alertable messages shipped with OS/400.

STRMSF and ENDMSF Command Error Recovery

When the STRMSF and ENDMSF commands are started:

Sometimes you may get a message indicating that MSF is already active. If this error occurs, you must end the mail server framework (using the ENDMSF command) before the MSF will start again.

What if the job log contains error message CPFABA4:

There are two ways to recover from the CPFABA4 error message.

- Use the Remove Mail Server Framework Configuration (QzmfRmvMailCfg) API.
- Use the Work with Registration display to remove an exit program. See Chapter 3, “Configuration Considerations” for more information about using this display.

Journal entries in the QZMF journal will help you determine when these errors occurred. See Appendix B, “MSF Journal Logging and Journal Formats” for more information about journal types and their descriptions.

Chapter 3. Configuration Considerations

This chapter explains how you can find the registered exit points and how to register exit programs with an exit point.

Exit Point Program Registration

An exit point program is registered with one of the following ten MSF exit points:

```

QIBM_QZMFMSF_LST_EXP
QIBM_QZMFMSF_ADR_RSL
QIBM_QZMFMSF_ENL_PSS
QIBM_QZMFMSF_ATT_CNV
QIBM_QZMFMSF_SEC_AUT
QIBM_QZMFMSF_LCL_DEL
QIBM_QZMFMSF_MSJ_FWD
QIBM_QZMFMSF_NON_DEL
QIBM_QZMFMSF_ATT_MGT
QIBM_QZMFMSF_ACT
  
```

To view the exit points, use the WRKREGINF command. You probably have a number of exit points registered on your system, but you can work with only MSF exit points by typing WRKREGINF FORMAT(MSFF0100) on the command line.

Exit points are shown in alphabetical order as follows:

```

Work with Registration Information

Type options, press Enter.
5=Display exit point 8=Work with exit programs

  Opt  Exit Point      Exit Point      Registered Text
  ---  ---            ---            ---
  ---  QIBM_QZMFMSF_ACT  MSFF0100      *YES  MSF Accounting Exit
  ---  QIBM_QZMFMSF_ADR_RSL  MSFF0100      *YES  MSF Address Resolution
  ---  QIBM_QZMFMSF_ATT_CNV  MSFF0100      *YES  MSF Attachment Conversion
  ---  QIBM_QZMFMSF_ATT_MGT  MSFF0100      *YES  MSF Attachment Management
  ---  QIBM_QZMFMSF_ENL_PSS  MSFF0100      *YES  MSF Envelope Processing
  ---  QIBM_QZMFMSF_LCL_DEL  MSFF0100      *YES  MSF Local Delivery
  ---  QIBM_QZMFMSF_LST_EXP  MSFF0100      *YES  MSF List Expansion
  ---  QIBM_QZMFMSF_MSG_FWD  MSFF0100      *YES  MSF Message Forwarding
  ---  QIBM_QZMFMSF_NON_DEL  MSFF0100      *YES  MSF Non Delivery
  ---  QIBM_QZMFMSF_SEC_AUT  MSFF0100      *YES  MSF Security and Authority
  ---  QIBM_QZMFMSF_TRK_CHG  MSFF0100      *YES  MSF Track Mail Message Change
  ---                                     More...

Command
====>
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
  
```

Figure 3-1. Work with Registration Information - First Display

```

Work with Registration Information

Type options, press Enter.
5=Display exit point 8=Work with exit programs

  Opt  Exit Point      Exit Point      Registered Text
  ---  ---            ---            ---
  ---  QIBM_QZMFMSF_VLD_TYP  MSFF0100      *YES  MSF Validate Type

Command
====>
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
  
```

Figure 3-2. Work with Registration Information - Second Display

The first ten exit points you see here are those described in Chapter 4, "Mail Server Framework Structure."

There are two other exit points (QIBM_QZMFMSF_TRK_CHG and QIBM_QZMFMSF_VLD_TYP). These exit points can be called by MSF to start a program to track changes made to an MSF message and to validate information associated with an MSF message. For more information about how an MSF message changes, refer to "How the MSF Message Changes" on page 5-4. This information is useful when testing exit point programs.

To display information about the specific exit point, use option 5. To view and work with programs registered with a specific exit point, use option 8. The address resolution exit point is displayed by typing 8 next to exit point QIBM_QZMFMSF_ADR_RSL in the *Opt* field.

```

Work with Registration Information

Type options, press Enter.
5=Display exit point 8=Work with exit programs

  Opt  Exit Point      Exit Point      Registered Text
  ---  ---            ---            ---
  ---  QIBM_QZMFMSF_ACT      MSFF0100      *YES  MSF Accounting Exit
  8  QIBM_QZMFMSF_ADR_RSL  MSFF0100      *YES  MSF Address Resolution
  ---  QIBM_QZMFMSF_ATT_CNV  MSFF0100      *YES  MSF Attachment Conversion
  ---  QIBM_QZMFMSF_ATT_MGT  MSFF0100      *YES  MSF Attachment Management
  ---  QIBM_QZMFMSF_ENL_PSS  MSFF0100      *YES  MSF Envelope Processing
  ---  QIBM_QZMFMSF_LCL_DEL  MSFF0100      *YES  MSF Local Delivery
  ---  QIBM_QZMFMSF_LST_EXP  MSFF0100      *YES  MSF List Expansion
  ---  QIBM_QZMFMSF_MSG_FWD  MSFF0100      *YES  MSF Message Forwarding
  ---  QIBM_QZMFMSF_NON_DEL  MSFF0100      *YES  MSF Non Delivery
  ---  QIBM_QZMFMSF_SEC_AUT  MSFF0100      *YES  MSF Security and Authority
  ---  QIBM_QZMFMSF_TRK_CHG  MSFF0100      *YES  MSF Track Mail Message Change
  ---                                     More...

Command
====>
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel
  
```

Figure 3-3. Work with Registration Information Display

You can add, remove, display, or replace exit programs using the Work with Exit Programs display. Figure 3-4 shows that SNA distribution services (SNADS) has a single exit program preregistered with this exit point. The other exit program is provided by the framework.

Attention

Do not add exit program numbers that have exit program numbers in multiples of 100 (for example, 100 or 200). Multiples of 100 are reserved for IBM use. Do not move IBM-supplied exit programs to different positions, change the sequence of the exit programs, or change exit point program data.

(For more information about how SNADS uses MSF, see Appendix D, "How Mail Server Framework Works with SNADS, Object Distribution, and OfficeVision.")

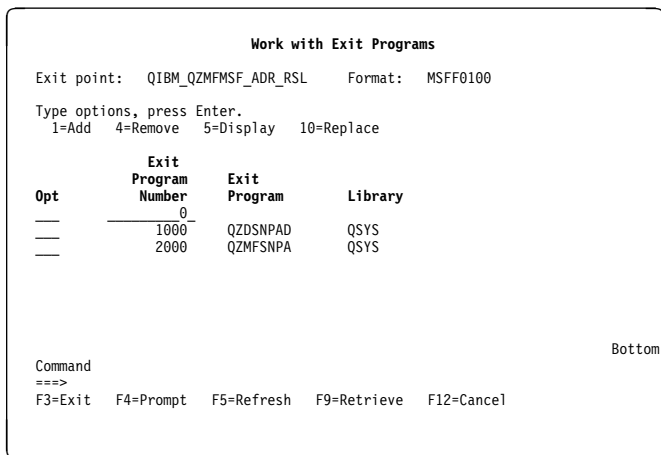


Figure 3-4. Work with Exit Programs Display

To display details about a specific exit point program, type 5 in the *Opt* field next to the exit program you want to display. The Display Exit Program display is shown.

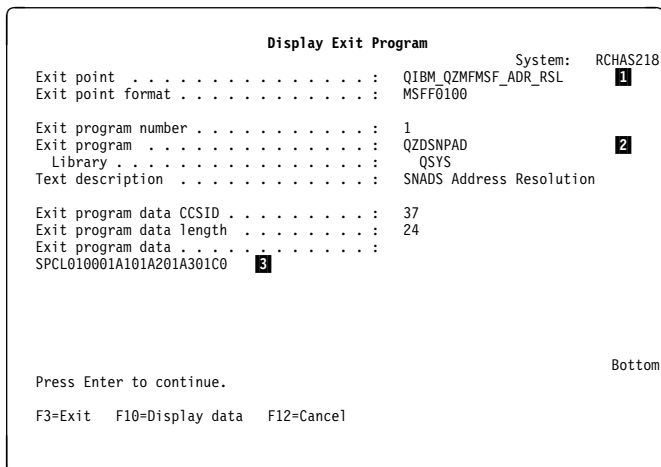


Figure 3-5. Display Exit Program Display

Figure 3-6 shows how the output on the display maps to the mail server framework.

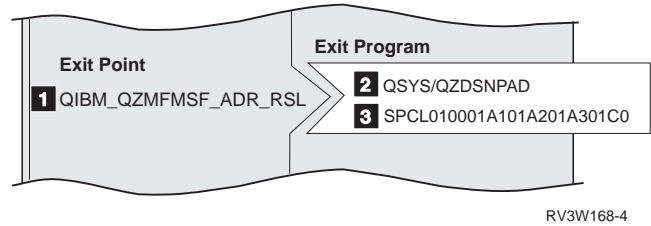


Figure 3-6. Example of an Exit Program in the Mail Server Framework

- 1 The exit point name, which is the address resolution exit point.
- 2 The name of the exit program, which is a program associated with the support of the SNADS function. See Appendix D, "How Mail Server Framework Works with SNADS, Object Distribution, and OfficeVision."
- 3 The exit program data used by the mail server framework to select the exit program (QSYS/QZDZNPAD). This data consists of a format name (SPCL0100), followed by MSF address types. These address types are used by the framework to determine if the exit program (QSYS/QZDZNPAD) should be called. Whenever an MSF message having addresses of these types in its recipient list is distributed by the framework, this exit program is called and passed information about the MSF message.

MSF Configuration APIs

The mail server framework requires that MSF data types be configured. These configured types are stored in internal files. There are APIs available to list, add, and remove the four data types from the framework. Appendix A, "Mail Server Framework APIs" on page A-1 contains a description of these APIs.

When an MSF message is created, the message information is assigned data types. The mail server framework validates the data types by looking in the internal table. If the data type is not configured, the MSF cannot process the message. Messages are logged in the job log of the program that attempted to use a data type that was not configured.

The same is true if an exit point program tries to change a message. If the data type is not configured, MSF cannot change the message. Messages are logged in the job log of the QMSF job in which the exit point program was called.

In either case, the owner of the failing program needs to look for the error in the exit point program. These instances are not indications of problems with the mail server framework.

It is important to note that when the program is changed, or any configuration changes are made using the MSF configuration APIs, you must *end* the mail server framework and

then *restart* the mail server framework for the changes to take effect.

See “MSF Data Types” on page 5-4 for more information about the four data types used by the mail server framework.

Chapter 4. Mail Server Framework Structure

The mail server framework provides the structure to support functions that do the work of electronic mail distribution. This chapter shows how an MSF message is processed by the framework.

Exit Points

An exit point is a specific point in a system function or program where control may be passed to one or more specified exit programs.

An exit point can call one program, a fixed number of programs, or all programs associated with an exit point. An exit point contains an exit point name and exit point format name.

Table 4-1. MSF Exit Points

MSF Exit Point Description	MSF Exit Point Name
List expansion	QIBM_QZMFMSF_LST_EXP
Address resolution	QIBM_QZMFMSF_ADR_RSL
Envelope processing	QIBM_QZMFMSF_ENL_PSS
Attachment conversion	QIBM_QZMFMSF_ATT_CNV
Security and authority	QIBM_QZMFMSF_SEC_AUT
Local delivery	QIBM_QZMFMSF_LCL_DEL
Message forwarding	QIBM_QZMFMSF_MSG_FWD
Non-delivery	QIBM_QZMFMSF_NON_DEL
Attachment management	QIBM_QZMFMSF_ATT_MGT
Accounting	QIBM_QZMFMSF_ACT

There are two other exit points, QIBM_QZMFMSF_TRK_CHG and QIBM_QZMFMSF_VLD_TYP.

“Exit Point Program Registration” on page 3-1 explains how to work with registration information about the exit points listed in Table 4-1.

See the *System API Reference* book for more information about exit points and exit point programs.

Exit Point Programs

An exit point program is a program to which control is passed from an exit point. Each exit point program is associated with an exit point and exit point format.

Snap-Ins

An MSF snap-in is an exit point program that is called from the mail server framework exit points (see Table 4-1). The mail server framework recognizes snap-ins as exit point programs.

The MSF message passes by the exit point if an exit point program is not configured for the exit point. The exit point program is called by the address types or message types in the message. More information about APIs can be found in Appendix A, “Mail Server Framework APIs.” For an example of how SNADS uses the mail server framework, see Appendix D, “How Mail Server Framework Works with SNADS, Object Distribution, and OfficeVision.”

Exit Points Provided on the Mail Server Framework

The MSF message enters at the top of the framework and continues to flow through the framework until it reaches the bottom.

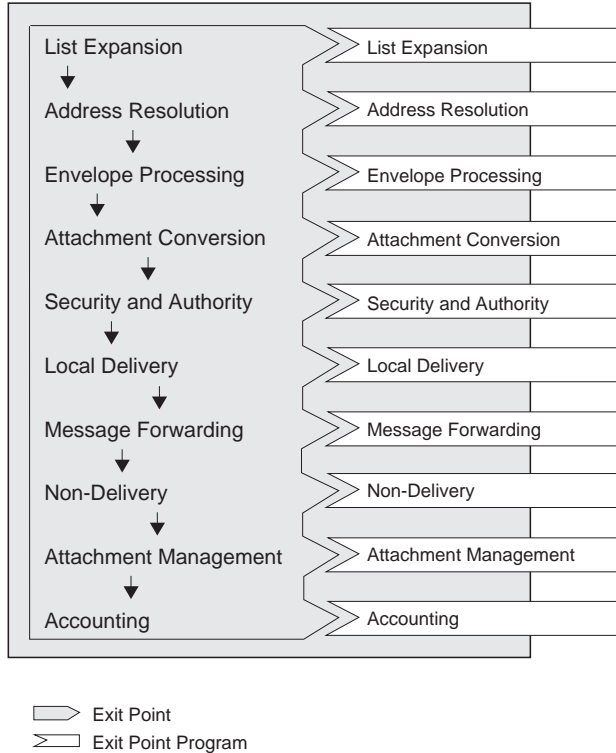


Figure 4-1. MSF Exit Point and Exit Point Program Overview

Figure 4-1 shows the exit points provided on the mail server framework. The MSF exit points provide a place for an exit program to be snapped in. A program that is snapped in at an exit point provides the function. These functions are not part of the mail server framework. The framework only provides the capability to call programs to correct, distribute, or log the MSF message information. The mail server framework uses the results of each of the exit point programs to determine which exit program to call next.

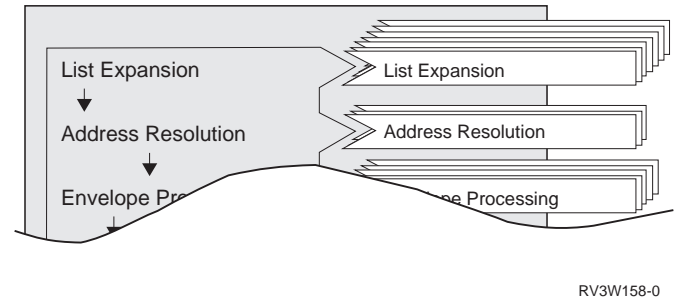
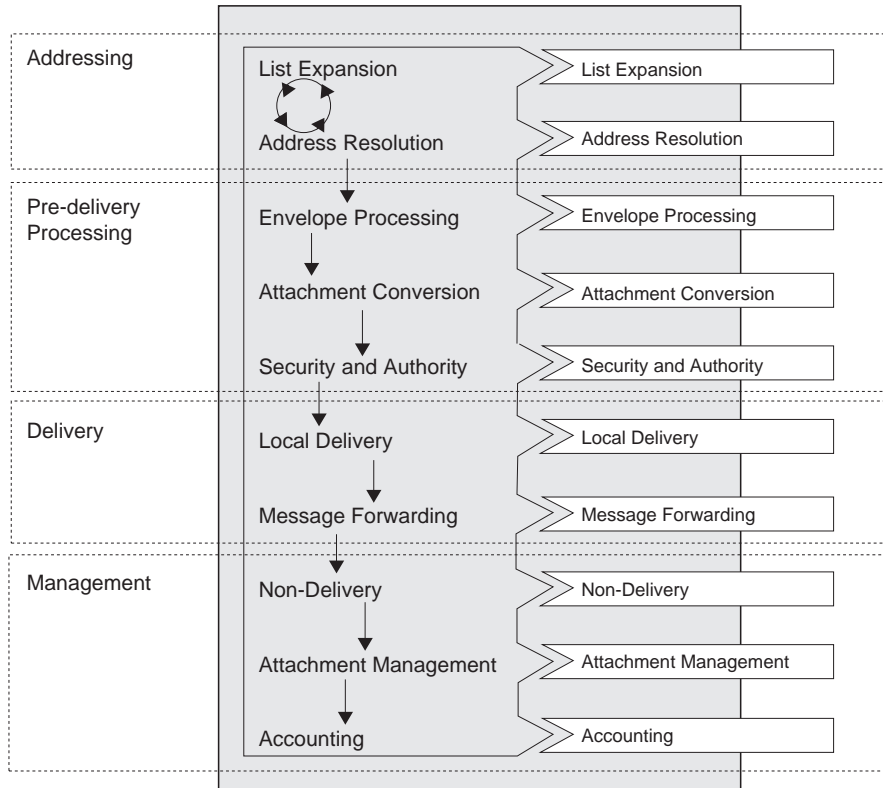




Figure 4-2. Using Multiple MSF Exit Point Programs

Multiple exit point programs can be configured for each mail server framework exit point. Figure 4-2 shows multiple exit point programs attached to an exit point.

MSF Exit Point Groups

It is easiest to think about the exit point programs in terms of groups of exit point programs that the exit points call. The exit points are described in groups that perform related processing functions.



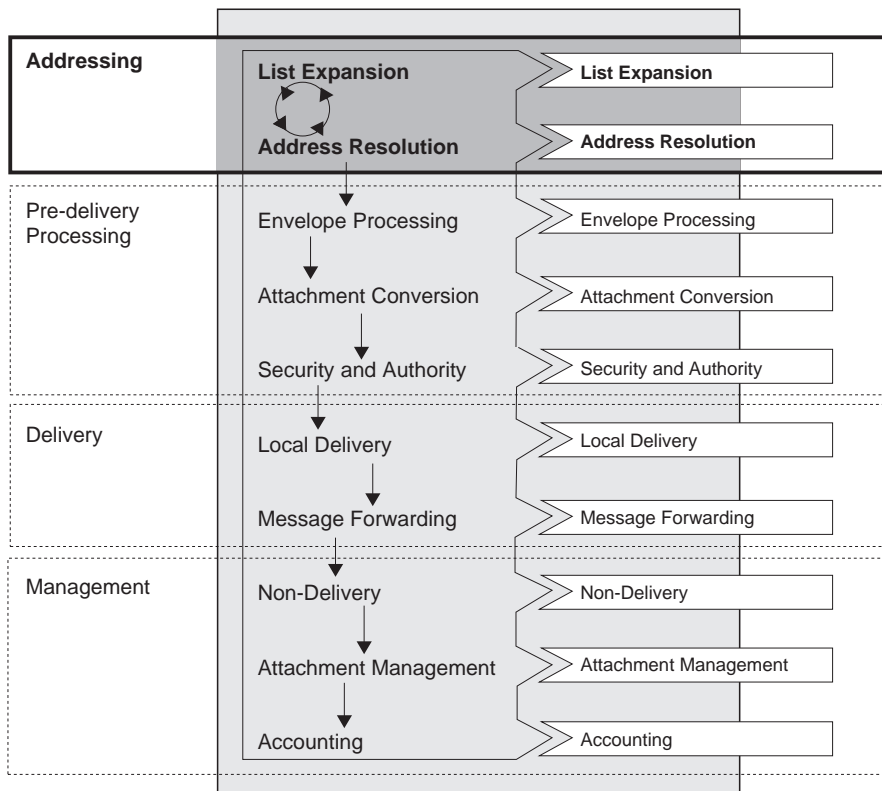
 Exit Point
 Exit Point Program

RV3W153-3

Figure 4-3. MSF Exit Point Groups

- Addressing group**
 The exit point programs in this exit point group resolve the addresses of the originator, recipient, report-to, and report-on lists.
 These exit point programs establish that every recipient has a MSF data type and that the recipient list is complete and correct. The mail server framework continues to call these exit point programs until the recipient list information is complete.
- Pre-delivery processing group**
 The pre-delivery processing group determines if the MSF data type needs to be changed for delivery to take place, including envelopes and attachment references.
- Delivery group**
 The delivery group exit point programs provide local delivery or forwarding of an MSF message.
- Management group**
 This is the clean-up area. The management group exit point programs determine what to do with the MSF message if it cannot be delivered or forwarded. It manages storage used by the MSF message attachments when the attachments are no longer needed. It can also log how and when the framework was used, track the processing of the MSF message, or provide accounting for the MSF message flowing through the framework.

Exit Point Programs in the Addressing Group



RV3W154-1

Figure 4-4. Exit Point Programs in the Addressing Group

List Expansion Exit Point

The list expansion exit point calls programs that determine if a message recipient address name is the name of a distribution list. The name of the distribution list can expand into either a list of recipients or more names of distribution lists. This exit point calls registered programs based on the address type of recipients that do not have a message type already assigned to them. A nondeliverable message type is assigned to those that do not have a message type already assigned. Every message recipient has an address and a type associated with the address. The list expansion exit point program is configured by address type. When an MSF message is being processed that contains recipients with that address type, then the program is called.

For example, a single recipient list address could represent the name of a distribution list containing the names of the members of a department. A list expansion exit point program function could expand the single recipient list address into the addresses of the department members.

To see how exit point programs are configured to be called, see Chapter 3, "Configuration Considerations."

Address Resolution Exit Point

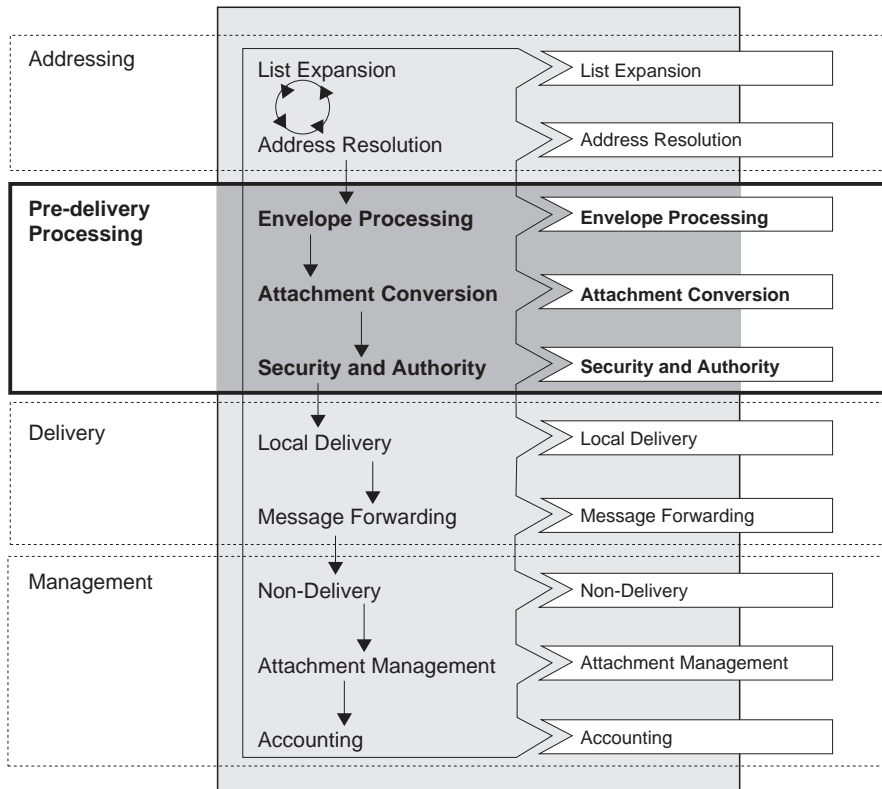
One of the main purposes of any program called by this exit point is to assign a message type and message status to every MSF message recipient. Recipient message types are used to select programs configured in the remaining exit points. It allows the user to configure an exit program that provides various routing functions. Sometimes the message originator does not provide all of the necessary distribution information. For example, the exit point program could take a partial recipient list, locate it in the directory, and then change the recipient information to include the complete information required by later exit point programs.

After all address resolution exit point programs are called, the mail server framework checks to see if a recipient list entry was replaced or added by a recipient without a message type. If this occurred, processing continues at the list expansion exit point, allowing the new information to be processed.

If any entries are added to the list either during list expansion or address resolution, the processing of the list expansion exit point program and the address resolution exit point program is recursive. Only the new entries are processed

because the other entries have already been processed.
See Figure 4-4.

Exit Point Programs in the Pre-Delivery Processing Group



RV3W155-2

Figure 4-5. Exit Point Programs in the Pre-Delivery Processing Group

Note

If a recipient list entry does not have a message data type at the start of the pre-processing delivery group, the recipient list entry is given a nondeliverable status and message type.

An envelope processing exit point program can add a new envelope type to the MSF message. If this occurs, the mail server framework does not call any of the exit point programs from previous exit points. An exit program might want to add another envelope if there is an exit point program later in the mail server framework processing that is designed to use that particular type of envelope.

Envelope Processing Exit Point

Any envelope processing program called by this exit point allows for message envelope alterations and conversions. An **envelope** is information associated with an MSF message and the recipient, such as subject, date and time, priority, notes, and special instructions.

This exit point program uses information about a recipient message type and envelope contents. It can provide message envelope updates, in addition to converting an envelope format into another envelope associated with a recipient message type.

Attachment Conversion Exit Point

The programs called by this exit point resolve potential conflicts between the previously resolved message type, and the format, structure, or content of the attachments associated with the MSF message. For example, a conflict could be that the attachments are not in the correct format, not stored in the correct file system, or need to be changed so that the exit point programs that follow can use the information. This exit point allows an attachment conversion exit point program to make the required changes to the message attachment content. For example, an OfficeVision document could be converted to simple ASCII text within this exit point program.

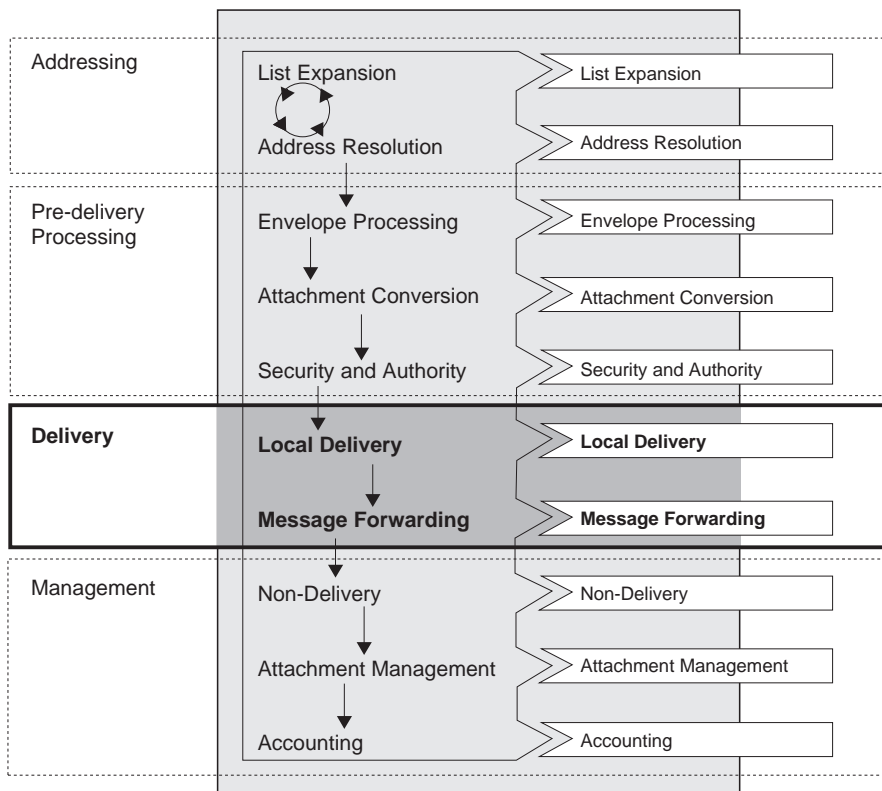
Security and Authority Exit Point

The security and authority exit point is available for users to write exit programs to do specific security and authority checking required by a system or mail server. An MSF message could be changed by the user-written exit program so all or some of its recipients would not receive the message information. To stop such a message, the recipient

distribution status for each refused recipient (for example, a remote recipient) could be changed to nondeliverable or security violation state by the program.

The exit program can access all information about the message (not the message itself) provided by the mail server framework.

Exit Point Programs in the Delivery Group



RV3W156-1

Figure 4-6. Exit Point Programs in the Delivery Group

Local Delivery Exit Point

The programs called by the local delivery exit point provide local delivery of the MSF message.

As with other MSF exit points, it is possible to configure exit point programs to call or pass recipients that have certain message types in a message recipient list. The message status of the recipient must be a local message status and the exit point program must be configured to handle recipients with that recipient message type.

Message Forwarding Exit Point

The programs called by the message forwarding exit point allow an MSF message to be forwarded to remote recipients (for example, a mail user not defined on the same AS/400 system).

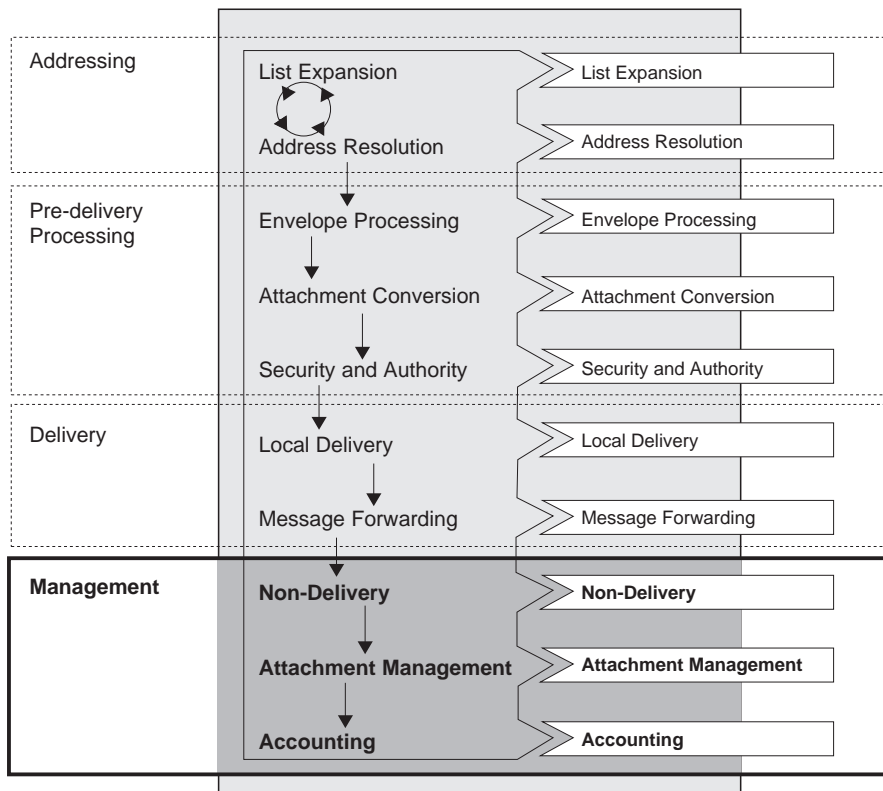
The mail server framework does not forward the message to

the message recipient. The framework only decides which exit point programs to call and pass the message information to. It uses the message types of recipient list entries that have a remote status to make these decisions.

As with other MSF exit points, it is possible to configure exit point programs that only call or pass recipients that have specific MSF data types assigned to them. The message status of the recipient must be remote and the exit point program must be written to handle recipients with that recipient message type.

Various exit point programs might be written to support protocol-specific message forwarding. For example, the SNADS message forwarding program forwards an MSF data type to remote SNADS recipients by creating a SNADS distribution and placing the distribution on a SNADS distribution queue. A SNADS sender will then do the actual sending of the distribution. See the *SNA Distribution Services* book for more information about SNADS distribution.

Exit Point Programs in the Management Group



RV3W157-1

Figure 4-7. Exit Point Program in the Management Group

Non-Delivery Exit Point

The programs called by the non-delivery exit point report that the MSF message was not delivered to its destination or recipient. The exit point program allows for the following solutions:

- Logging the distribution that cannot be delivered
- Routing the nondeliverable messages to a storage area where they are kept, corrected, and resent
- Creating a new message to report the non-delivery

Several things can prevent a message from being delivered. Examples include the following:

- An address resolution exit point program may have a partial address of a recipient. The exit point program cannot construct a complete address, and the MSF message is labeled nondeliverable.
- The local delivery exit point program or the message forwarding exit point program may be configured incorrectly.
- Recipients may have been marked nondeliverable through address resolution processing.

- A recipient does not have a message type and was marked nondeliverable by the mail server framework.

Attachment Management Exit Point

The programs called by the attachment management exit point manage the attachments to an MSF message. This exit point is used only if there is an attachment reference list associated with the MSF message.

MSF messages only refer to attachments to a message; they do not own or manage the storage of the attachments to the message.

Accounting Exit Point

The programs called by the accounting exit point support the system or network accounting functions required in the electronic mail environment. The accounting is done after all other activity has taken place in the other exit point programs. The user can write an exit program that records the activity that takes place in the mail server framework.

This is the last point at which message information is available. The mail server framework deletes the internal struc-

tures associated with the message as soon as the appropriate exit point programs have used the information. The MSF message is destroyed after this exit point. An exit

point program must be written and called to save, deliver, or forward any message information.

Chapter 5. Mail Server Framework Message Concepts

MSF messages are created when the Create Mail Message (QzmfCrtMailMsg) API is used. The framework passes information about the MSF message across the API. The MSF message is put on a queue. MSF messages exist until all of the appropriate exit point programs have processed them.

The MSF message contains **typed** information about the message. The mail server framework contains interfaces to exit points that deal with the MSF message.

The MSF message lists and data types allow the framework to work with the MSF message information. The MSF message can be used or changed only by an exit point program when the framework calls it and passes the MSF message information to the program.

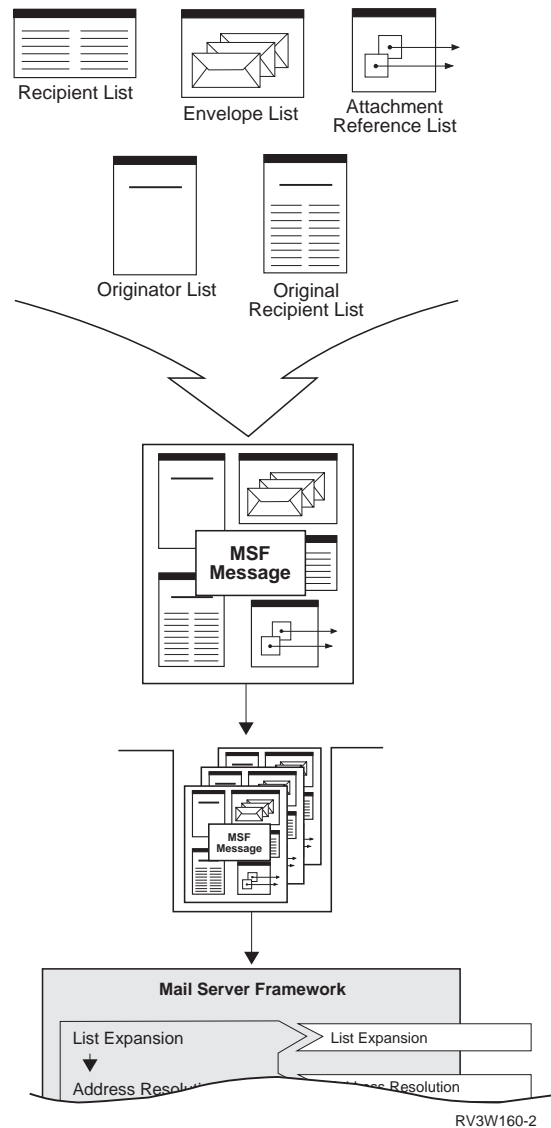


Figure 5-1. MSF Message Concept Overview

MSF Message Lists

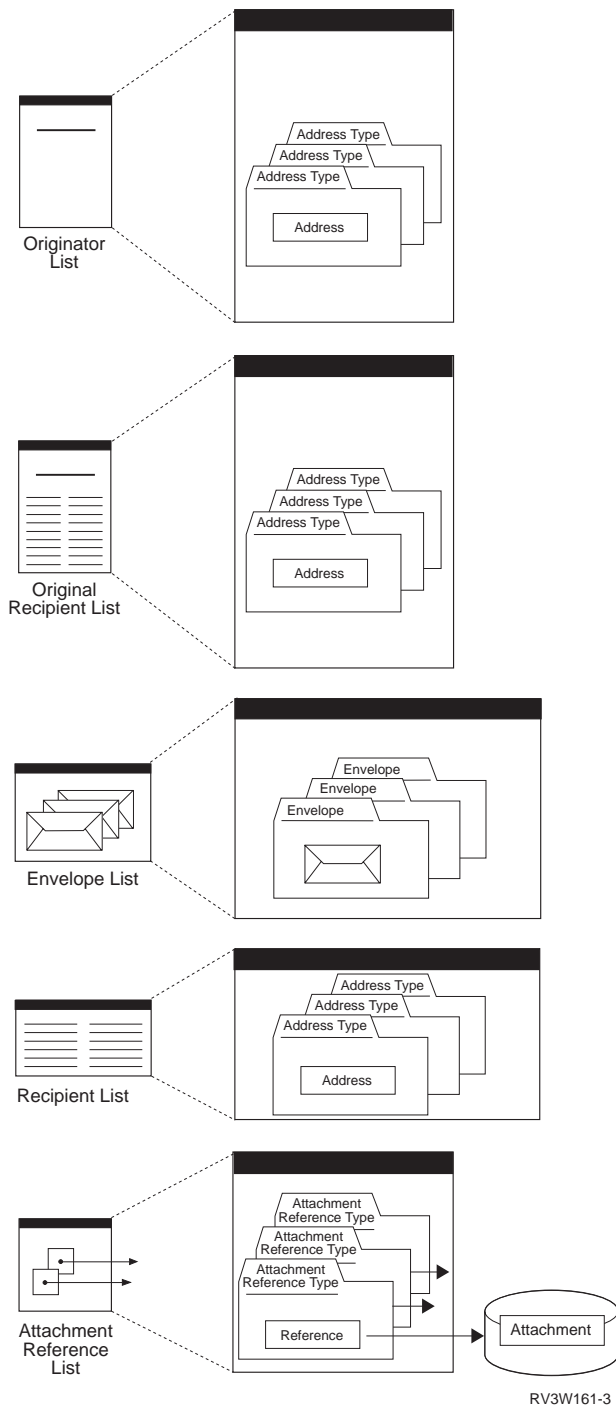
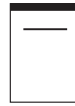


Figure 5-2. MSF Message Lists

Originator List



The MSF message must have an originator list with at least one originator entry. Each originator entry is made up of an originator address and an originator address type. If more than one originating entry is specified when the message is created, the mail server framework assumes that all of the entries represent the same originator.

Original Recipient List



An MSF message can contain a list of all the recipients that were originally specified when the message was sent. This optional list can be used by applications that, for example, want to reply to all recipients of a message. This list is different from the recipient list in that, as a message moves through the network, recipients that are delivered on intermediate systems are removed from the recipient list. However, the original recipient list remains intact.

Recipient List



An MSF message must have one or more recipient entries. When an MSF message is created, the recipient address and recipient address type are required fields for each recipient entry.

While the message type in the recipient list entry is not a required field for creating a message, it must be assigned to a recipient by one of the addressing exit point programs. The MSF data type assigned must be in the table of acceptable MSF data types for the system.

The key parts of a recipient list entry are:

- Address types
- Message types
- Status field

Envelope List



An MSF message must have one or more envelope entries. The message envelope is the information about a particular message for a particular protocol type, except for the generic envelope type.

A generic envelope type is defined for any exit point program to use. Using the generic envelope type allows an exit point program that is processing messages for one protocol type to hand over the message to another exit point program that processes messages for another protocol type.

The mail server framework requires that each envelope entry contain envelope information and an identifying envelope type. The envelope type must be in the table of acceptable protocol types for the system.

Attachment Reference List



An MSF message can refer to any number of attachments. Each attachment referenced is an entry in the attachment reference list. The MSF messages without attachments have no attachment reference list.

The mail server framework requires that each attachment reference list contain attachment reference data and a reference type. The reference type must be in the table of

acceptable reference types for the system. For example, a reference type can be a database file member or a folder document.

Other Lists

There are other lists that can be passed to the Create Mail Message API.

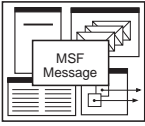
Report-On List: The mail server framework requires that each report-on list entry contain a non-delivery address and a corresponding address type. The address type must be configured as a valid address type in the mail server framework by using the configuration APIs described in “MSF Configuration APIs” on page A-1.

Report-To List: The report-to list contains a list of users that are sent an error report when an error occurs with a message. The mail server framework requires that each report-to address entry contain a report-to address and a corresponding address type. An MSF message can have zero, one, or multiple report-to address entries. The address type must be configured as a valid address type in the mail server framework by using the configuration APIs described in “MSF Configuration APIs” on page A-1.

Message Identifier

The message identifier (ID) is a 32-character unique ID generated by the mail server framework when the message is created. This ID is used by the exit point programs when they are called to retrieve or change information in the MSF message. The message identifier cannot be reused. The ID is kept with the message and passed to all of the exit point programs.

MSF Data Types



The MSF data type is used by the mail server framework to define the contents of the data and how the data should be processed by the exit point programs. The MSF does not process the contents of the message. The framework passes the MSF message to the exit point programs for processing.

MSF data types are used in the exit program data when configuring exit point programs. Figure 3-6 on page 3-2 shows an example of exit program data **3**.

The MSF **data type** determines which exit point program processes the message and how the MSF message is processed. Figure 5-3 on page 5-5 shows the framework using MSF data types to determine which exit point program to call.

There are four data types that are defined by the mail server framework.

- Address type

The address type is an identifier that indicates the type of data that is contained in the address string. The address type determines what exit point program to call for the list expansion and address resolution exit points.

- Message type

The message type is used in the remaining exit points to identify which exit point program to call.

- Envelope type

The envelope type is an identifier that indicates the type of data that is contained in the envelope.

- Attachment reference type

The attachment reference type is an identifier that indicates the type of data that is contained in the attachment reference.

Rules for MSF Data Type Definitions

Every MSF data type definition is associated with a type group, consisting of a data type value, type name, and type text. There are IBM-supplied data types predefined in the MSF configuration that are shipped with the OS/400 licensed program. See *AnyMail/400 Mail Server Framework Developer Guide* for more information about the IBM-supplied data types.

The **data type value** is a unique 4-character representation. The following type values have special meaning to the mail server framework.

- 0xxx to 9xxx are reserved for use by IBM.

Important

It is not recommended to define a data type using this range. IBM may, at a later time, add additional predefined data types within this range. If a data type definition is found in this range and it is not supplied by IBM, problems could occur for MSF mail applications.

- 9998 is reserved for use as the nondelivery message data type.
- 9999 is not a data type. It is reserved for use in specifying that an exit point program is configured to process all address or message data types.

The **type name** is the name of the data type being configured. The **type text** is a description of the data type definition. Figure B-12 on page B-11 is an example of a CF journal entry, showing the type group, data type value, and type name.

How the MSF Data Types Are Used to Call Exit Point Programs

Exit point programs are called by the mail server framework based on the following:

- MSF data types specified in the exit program data of the registered exit point program.
- MSF data types present in the MSF message recipient list entries.

An exit point program is called and passed information about the MSF message so that the program can perform its functions.

How the MSF Message Changes

Exit point programs that are called by the mail server framework can change the information in an MSF message so other exit point programs can see different or additional message information. An exit point program can use the Change Mail Message (QzmfChgMailMsg) API to make changes to the MSF message. These changes have no effect until the program returns control to the framework, indicating that it has finished processing the MSF message successfully. See Figure 5-4 on page 5-5.

There are some restrictions regarding which exit point programs may change an MSF message at certain exit points.

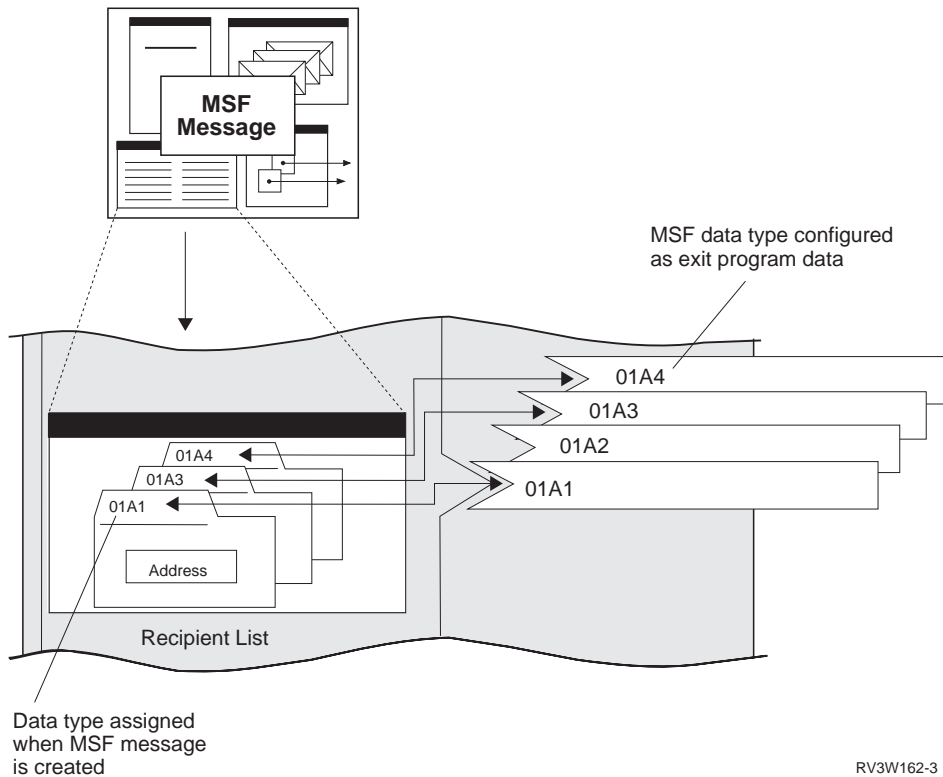


Figure 5-3. MSF Data Types Used to Determine Which Exit Point Program to Call

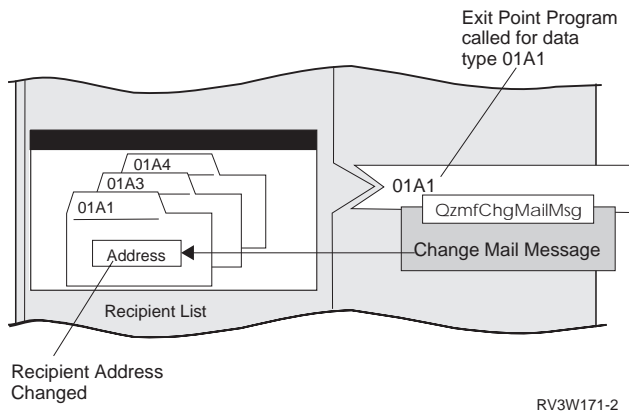


Figure 5-4. Exit Point Programs Can Change Information in an MSF Message

How the MSF Message Flows Through the Framework

A user mail application on the system uses the mail server framework Create Mail Message API to create MSF messages. The mail server framework inspects the MSF messages to ensure that the information is acceptable.

Once MSF messages are created, they are put on a queue. When the mail server framework has been started by the Start Mail Server Framework (STRMSF) command, the MSF message is processed by the framework. Processing of the MSF message is handled by the QMSF job (located in the QSYSWRK subsystem). See “Managing MSF Jobs” on

page 2-2 for more information about the QSYSWRK subsystem.

The mail server framework decides which exit point programs to call at each exit point. The exit point programs can act on or change message information that might be used by other exit point programs. This does not mean that each message uses every registered exit point program; the message data types of the recipient determine which exit point programs are called.

It also does not mean that every exit point program called uses the MSF message information. The exit point program determines that it has no function to perform for that message and simply returns it to the framework.

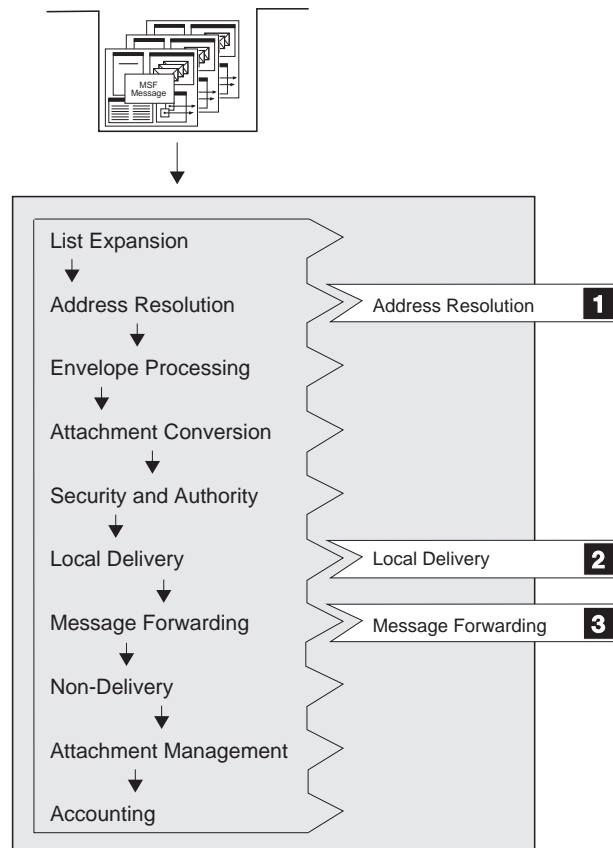
In Figure 5-5 on page 5-6, the address resolution exit point program is the first program called **1** based on the MSF address types in the message recipient list. This exit point program decides which MSF message types and recipient status are assigned to recipients in the message recipient list. It changes the recipient list entries. In this example, some message recipients are assigned local status and some are assigned remote status.

A local delivery exit point program **2** is called by the framework and passed information in the MSF message. The information includes the recipient list entries that have the message type (and local status) that the exit point program is configured to handle.

When the program returns, a message forwarding exit point program **3** is called and passed information in the MSF

message. The information includes the recipient list entries that have the message type (and forward status) that the exit point program was configured to handle. When the message forwarding program returns, the framework determines if other exit point programs at the remaining exit points should be called based on the message types in the message recipient list. In this example, there are no additional exit point programs to be called so the MSF message is removed.

Figure 5-5 shows three exit point programs cooperating in delivering MSF message information to local recipients and forwarding that information to remote recipients in its recipient list. There could be more exit point programs configured but not part of the example message flow because the MSF data types did not result in a call to these exit point programs.



RV3W169-0

Figure 5-5. How an MSF Message Flows Through the Framework

Appendix A. Mail Server Framework APIs

This appendix describes the mail server framework APIs. For more information about the required parameter groups, message descriptors, message descriptor formats, and error messages, see the *System API Reference* book.

The following APIs are made up of program calls:

MSF configuration APIs

- Add Mail Server Framework Configuration (QzmfAddMailCfg)
- List Mail Server Framework Configuration (QzmfLstMailCfg)
- Remove Mail Server Framework Configuration (QzmfRmvMailCfg)

Message APIs

- Create Mail Message (QzmfCrtMailMsg)
- Change Mail Message (QzmfChgMailMsg)
- Retrieve Mail Message (QzmfRtvMailMsg)
- Complete Creation Sequence (QzmfCrtCmpMailMsg)
- Query Mail Message Identifier (QzmfQryMailMsgId)
- Reserve Mail Message Identifier (QzmfRsvMailMsgId)
- Remove Reserved Mail Message Identifier (QzmfRmvRsvMailMsgId)

The mail server framework also provides three exit points where the following exit programs are used:

- Snap-In Call Exit Point Program
- Track Mail Message Changes Exit Point Program
- Validate Data Field Exit Point Program

MSF Configuration APIs

The MSF configuration APIs are used to process type configurations within the mail server framework configuration. The following processes can be performed:

- Adding to the configuration
- Removing from the configuration
- Listing the contents of the configuration

These APIs are used to add or remove new mail application support in the framework or to list the current configuration.

Exit point programs can also access the trace information that contains an account of all the exit point programs used in the framework and those that changed the message by using the Retrieve Mail Message (QzmfRtvMailMsg) API. The framework does not record the exact changes made to the message. The Track Mail Message Changes Exit Program is called whenever an exit point program changes the message and a program is registered for the exit point.

Add Mail Server Framework Configuration (QzmfAddMailCfg) API

The Add Mail Server Framework Configuration (QzmfAddMailCfg) API configures the MSF data types used by the mail server framework to process messages. The following data types are valid:

- Address
- Message
- Envelope
- Attachment reference

This API also configures the address type and message type that can be specified as exit program data in the user-written exit program.

The MSF data type is used to process the MSF message through the framework. Only one data type can be added at a time. See "MSF Data Types" on page 5-4 for more information about the four data types.

List Mail Server Framework Configuration (QzmfLstMailCfg) API

The List Mail Server Framework Configuration (QzmfLstMailCfg) API creates a list of data types and places the list in a specified user space.

Remove Mail Server Framework Configuration (QzmfRmvMailCfg) API

The Remove Mail Server Framework Configuration (QzmfRmvMailCfg) API removes MSF data types that the mail server framework uses to process MSF messages. The MSF data type is removed from the configuration when this API is used. Only one data type can be removed at a time. See "MSF Data Types" on page 5-4 for more information about the four data types.

MSF Message APIs

The MSF message APIs are used to create an MSF message and ensure delivery of the message. These APIs are used by user-written exit programs.

Create Mail Message (QzmfCrtMailMsg) API

The Create Mail Message (QzmfCrtMailMsg) API creates an MSF message. When an MSF message is created, the following lists can be specified:

- Originator (required)
- Recipient (required)
- Envelope (required)

- Attachment reference
- Report-on
- Report-to
- Original recipient list
- Reply-to list

The format of the data is defined by assigning a *type* to the information. The MSF data types need to be configured before they can be supported. There are four MSF data types.

- Address type
- Message type
- Envelope type
- Attachment reference type

See Chapter 5, “Mail Server Framework Message Concepts” for more information about MSF lists and MSF data types.

Change Mail Message (QzmfChgMailMsg) API

The Change Mail Message (QzmfChgMailMsg) API changes information about an MSF message that was previously created using the Create Mail Message API. Each time it is called, the Change Mail Message API can change one or more list items for each of the message parameter list formats. For example, one call can result in changes to the envelope list, the recipient list, and the attachment reference list. An exit program can be written to change an MSF message. If the requested changes are acceptable, the existing list items are replaced with the new items that are specified. The list items do not retain their same unique list identifiers after the changes are complete.

Figure 5-4 on page 5-5 shows an example of how the MSF message changes when this API is used.

Note: The Change Mail Message API is only valid within the processing of a Snap-In Call Exit Point Program.

Retrieve Mail Message (QzmfRtvMailMsg) API

The Retrieve Mail Message (QzmfRtvMailMsg) API retrieves information about an MSF message and returns it in the receiver variable provided by the caller.

Note: The Retrieve Mail Message API is only valid within the processing of a Snap-In Call Exit Point Program or the Track Mail Message Changes Exit Point Program.

Other Optional APIs

Query Mail Message Identifier (QzmfQryMailMsgId) API

The Query Mail Message Identifier (QzmfQryMailMsgId) API is used to query the status of a mail message identifier within the mail server framework.

Reserve Mail Message Identifier

(QzmfRsvMailMsgId) API: The Reserve Mail Message Identifier (QzmfRsvMailMsgId) API is used to reserve an identifier for an electronic mail message.

Complete Creation Sequence

(QzmfCrtCmpMailMsg) API: The Complete Creation Sequence (QzmfCrtCmpMailMsg) API removes a previously created, reserved mail message identifier from the MSF list of reserved identifiers and acknowledges that the MSF message was created.

Remove Reserved Mail Message Identifier

(QzmfRmvRsvMailMsgId) API: The Remove Reserved Mail Message Identifier (QzmfRmvRsvMailMsgId) API removes a reserved identifier for an electronic mail message that you have not yet created. If the message was created, you must use the Complete Creation Sequence API to release the reserved message. After the Remove Reserved Mail Message Identifier API completes successfully, you cannot use the message identifier you reserved earlier when the message was created.

Exit Points

The mail server framework provides three exit points where the following exit programs are used.

Snap-In Call Exit Point Program

The Snap-In Call Exit Point Programs process the electronic mail within the framework.

Track Mail Message Changes Exit Point Program

In some cases, the user may want to track the changes made to a message. The Track Mail Message Changes Exit Program can be used. Whenever a message is changed by an exit point program, an exit program registered for this exit point is called and the following information is passed:

- The message identifier
- The exit point name being processed when the change was made
- The exit point program name responsible for the change

Validate Data Field Exit Point Program

The Validate Data Field Exit Point Program allows exit programs to validate the data for a message based on the MSF data type selected. As part of the mail server framework configuration, MSF data types must be defined to the system.

Appendix B. MSF Journal Logging and Journal Formats

This appendix provides information about the journal support used by the mail server framework.

QZMF Journal

The mail server framework uses OS/400 journal support to track MSF configuration changes and MSF message processing performed on the local system by the mail server framework.

A journal (QZMF) is shipped with security officer authority in the QUSRSYS library. The journal name used for the mail server framework must be QZMF. QZMF uses a journal code of S. Code S is a distributed mail service for SNA distribution services (SNADS), network alerts, or the mail server framework. For the QZMF journal, this code always means the mail server framework.

There are four types of MSF journal entries:

- LG MSF message log information for the mail server framework. A normal function, such as creating or completing an MSF message, was successfully performed.
- ER MSF message error information for specific MSF messages in the mail server framework. The MSF message could not be completed because of a framework or exit point program error.
- SY MSF system information for the mail server framework.
- CF Configuration information for the mail server framework.

An output file for each type of entry is provided that shows the data for each entry. This file enables you to copy the information to an output file using the Display Journal (DSPJRN) command. Specify *TYPE1 for the outfile format (OUTFILFMT) parameter.

You are responsible for changing the QZMF journal receiver with the Change Journal (CHGJRN) command. You can do this when the receiver is full, or at any convenient time.

When an MSF message is created or processed by the mail server framework successfully, a journal entry, type LG, is made on the system. If an error occurs when processing an MSF message, an error entry, type ER, is made in the QZMF journal. SY entries record significant events that may affect MSF functions, for example when the STRMSF or ENDMSF commands are used. An entry is made whenever a config-

uration of MSF types or exit programs takes place; this entry is type CF.

Displaying the QZMF Journal

You can view the MSF journal entries by using the Display Journal (DSPJRN) command. Type DSPJRN on the command line and press F4. You are prompted for the information you want to see. You can enter specific parameters and options or take the default values for each of the prompted parameters.

```

                                Display Journal (DSPJRN)
Type choices, press Enter.
Journal . . . . . QZMF      Name, *INTSYSJRN
Library . . . . . *LIBL     Name, *LIBL, *CURLIB
Journalled physical file:
File . . . . . *ALLFILE    Name, *ALLFILE, *ALL
Library . . . . .           Name, *LIBL, *CURLIB
Member . . . . .           Name, *FIRST, *ALL
+ for more values
Range of journal receivers:
Starting journal receiver . . *CURRENT  Name, *CURRENT, *CURCHAIN
Library . . . . .           Name, *LIBL, *CURLIB
Ending journal receiver . . . Name, *CURRENT
Library . . . . .           Name, *LIBL, *CURLIB
Starting sequence number . . . *FIRST   Number, *FIRST
Starting date and time:
Starting date . . . . . 01/16/95  Date
Starting time . . . . . 08:00:00  Time
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure B-1. Display Journal - First Display

```

                                Display Journal (DSPJRN)
Type choices, press Enter.
Ending sequence number . . . . *LAST     Number, *LAST
Ending date and time:
Ending date . . . . . 01/16/95  Date
Ending time . . . . .           Time
Number of journal entries . . . *ALL     Number, *ALL
Journal codes:
Journal code value . . . . . S      *ALL, *CTL, A, C, F, J, L...
Journal code selection . . . .     *ALLSLT, *IGNFILSLT
+ for more values
Journal entry types . . . . . *ALL     Character value, *ALL, *RCD
+ for more values
Job name . . . . . *ALL           Name, *ALL
User . . . . .           Name
Number . . . . .           000000-999999
Program . . . . . *ALL           Name, *ALL
User profile . . . . . *ALL           Name, *ALL
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure B-2. Display Journal - Second Display

```

Display Journal (DSPJRN)

Type choices, press Enter.

Commit cycle identifier . . . . *ALL      Number, *ALL
Dependent entries . . . . . *ALL      *ALL, *NONE
Output format . . . . . *CHAR      *CHAR, *HEX
Output . . . . . *          *, *PRINT, *OUTFILE

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure B-3. Display Journal - Third Display

The following specifies the parameters and the options available for MSF entries.

- **Range of journal receivers:** Specifies the starting (first) and ending (last) journal receivers that contain the journal entries being converted for output.
- **Starting sequence number.** Specifies the first journal entry to be considered for conversion for output. You can specify *FIRST to display the entire journal receiver range or you can specify a specific sequence number to be converted.
- **Starting date and time:** Specifies the date and time of the first journal entry.

- Starting date: The format for the date is defined by the job attributes DATFMT and, if separators are used, DATSEP (for example, 01/16/95).
- Starting time: The time is specified in 24-hour format and can be specified with or without a time separator (for example, 08:00:00).

These fields are very useful because they segment the QMSF journal entries into specific windows of interest. For example, if you want to display all entries that occurred on 05/24/95, starting at 11:00 a.m., specify the starting date of 05/24/95 and the starting time of 11:00:00, and then press the Enter key.

- **Ending sequence number.** Specifies the last journal entry to be considered for conversion. You can specify *LAST to display the final journal receiver being converted or you can specify a specific sequence number to be converted.
- **Ending date and time:** Specifies the date and time of the last journal entry.
 - Ending date: The format for the date is defined by the job attributes DATFMT and, if separators are used, DATSEP (for example, 05/30/95).

- Ending time: The time is specified in 24-hour format and can be specified with or without a time separator (for example, 08:00:00).

- **Journal codes:** For the mail server framework entries, this code will always be S. No other codes are used by MSF.
- **Journal entry types:** For the mail server framework, the following entries are used:
 - LG: MSF message log information entries
 - ER: MSF message error information entries
 - SY: MSF message system information entries
 - CF: MSF configuration information entries

You can specify any combination of codes. For example, you can specify that you only want to see the LG and ER entries. The default displays all entries.

- **Job name:** Specify QMSF or the user job name (for example, QPADEV0007) to show the jobs running for the mail server framework.

Displaying the Journal Entries

From the Display Journal display, press the Enter key to see the Display Journal Entries display. The information contained on this display depends on the parameters and values you specified (including default values taken). The entries are always displayed in chronological order. The following is an example of the display:

```

Display Journal Entries

Journal . . . . . : QZMF      Library . . . . . : QUSRSYS

Type options, press Enter.
5=Display entire entry

Opt  Sequence  Code  Type  Object  Library  Job  Time
-----
1    J    PR    QZMF  QUSRSYS  DSP01   19:15:28
3    J    RR    QZMF  QUSRSYS   19:42:51
4    J    IN    QZMF  QUSRSYS   22:21:06
5    J    IN    QZMF  QUSRSYS   11:43:28
6    S    SY    QZMF  QUSRSYS  QPADEV0007 12:43:31
7    S    SY    QZMF  QUSRSYS  QPADEV0007 13:01:42
8    S    SY    QZMF  QUSRSYS  QPADEV0008 14:10:22
9    S    SY    QZMF  QUSRSYS  QPADEV0008 14:10:31
10   S    SY    QZMF  QUSRSYS  QPADEV0008 14:10:33
11   S    SY    QZMF  QUSRSYS  QPADEV0008 14:13:27
12   S    SY    QZMF  QUSRSYS  QPADEV0008 14:13:34
13   S    SY    QZMF  QUSRSYS  QPADEV0008 14:13:35 +

F3=Exit  F12=Cancel

```

Figure B-4. Display Journal Entries Display

This sample display was shown with default values specified for all of the parameters. As a result, there are different codes, different entry types, and different jobs listed.

If the data in the record does not fit on one display, use the page keys to advance to subsequent displays or to return to a previous display in the series. A plus sign (+) in the lower right corner of the display indicates that there are more displays in the series.

Note: If there are no entries in the journal receiver that match the values on the entered parameters, this message appears on the display:
(No log entries).

This condition can occur when the log entries are sent to the journal receiver during a time when the system clock is set to an incorrect date or time. You would be unable to find entries for a specified range when you use a range search to search the affected journal receiver. Use the Change Journal (CHGJRN) command to create a new journal receiver. Creating a new journal receiver (once the system clock has been set to the correct time) ensures that the current journal receiver entries have the correct time. This command has no effect on old journal receivers.

From the Display Journal Entries display, you can type a 5 (Display entire entry) in the *Opt* field to see the detail for each entry. The information available in the detail is different for each entry type, so the detail has a unique display based on the entry type. The following are examples of detail displays for different function and entry types.

Entry Type LG: This display is shown when you type a 5 (Display entire entry) on the Display Journal Entries display next to an entry that has function type LG (logging). "Format for MSF Message Logging (LG)" on page B-5 provides detailed information about the contents of this display.

```

Display Journal Entry

Object . . . . . : Library . . . . . :
Member . . . . . : Sequence . . . . . : 3380
Code . . . . . : S - Distributed mail services
Type . . . . . : LG - Mail logging table information

Entry specific data
Column *...+...1...+...2...+...3...+...4...+...5
00001 'QMSF QMSF 006000QZMFBIGE2ID 100003394051'
00051 '52158420000000009 '

Press Enter to continue.

F3=Exit F6=Display only entry specific data
F10=Display only entry details F12=Cancel F24=More keys

```

Figure B-5. Display Journal Entry - Type LG

Entry Type ER: This display is shown when you type a 5 (Display entire entry) on the Display Journal Entries display next to an entry that has function type ER (error). "Format for MSF Message Errors (ER)" on page B-7 provides detailed information about the contents of this display.

```

Display Journal Entry

Object . . . . . : Library . . . . . :
Member . . . . . : Sequence . . . . . : 3395
Code . . . . . : S - Distributed mail services
Type . . . . . : ER - Mail error information

Entry specific data
Column *...+...1...+...2...+...3...+...4...+...5
00001 'QMSF QMSF 006000QZMFBIGE2ID 100003394051'
00051 '61305090000000015 CAPITST DEBPGM '

Press Enter to continue.

F3=Exit F6=Display only entry specific data
F10=Display only entry details F12=Cancel F24=More keys

```

Figure B-6. Display Journal Entry - Type ER

Entry Type SY: This display is shown when you type a 5 (Display entire entry) on the Display Journal Entries display next to an entry that has function type SY (system). "Format for MSF System Level Events Table (SY)" on page B-8 provides detailed information about the contents of this display.

```

Display Journal Entry

Object . . . . . : Library . . . . . :
Member . . . . . : Sequence . . . . . : 3715
Code . . . . . : S - Distributed mail services
Type . . . . . : SY - Mail system information

Entry specific data
Column *...+...1...+...2...+...3...+...4...+...5
00001 'QPADEV0007BANER 006163QZMFEEND5 '

Press Enter to continue.

F3=Exit F6=Display only entry specific data
F10=Display only entry details F12=Cancel F24=More keys

```

Figure B-7. Display Journal Entry - Type SY

Entry Type CF: This display is shown when you type a 5 (Display entire entry) on the Display Journal Entries display next to an entry that has the entry type CF (configuration). “Format for MSF Configuration Changes (CF)” on page B-9 provides detailed information about the contents of this display.

```

                                     Display Journal Entry
Object . . . . . :                               Library . . . . . :
Member . . . . . :                               Sequence . . . . . : 3690
Code . . . . . : S - Distributed mail services
Type . . . . . : CF - Mail configuration information

      Entry specific data
Column *...+...1...+...2...+...3...+...4...+...5
00001 'QPADEV0007BANER      006163QZMFDLTC4      0101TT01T'
00051 'TNAME'

                                     Bottom

Press Enter to continue.

F3=Exit  F6=Display only entry specific data
F10=Display only entry details  F12=Cancel  F24=More keys

```

Figure B-8. Display Journal Entry - Type CF

Format for MSF Message Logging (LG)

The entry is made when a MSF message is created, reset, or completed. The entry is mapped by the database file record, QAZMFLG, that represents the MSF message log information entered. This record is defined by the physical file QAZMFLG, which is shipped in QSYS library. The LG-type journal entry contains the following information:

Table B-1. Type LG Journal Entries for MSF Messages

Field	Format	Description
Entry length	Zoned(5,0)	Total length of the journal entry, including the entry length field.
Sequence number	Zoned(10,0)	Applied to each journal entry. Initially set to 1 for each new or restored journal. Reset when a new receiver is attached.
Journal code	Char(1)	Always S for MSF entries.
Entry type	Char(2)	Always LG for MSF message entries.
Date stamp	Char(6)	The system date that the entry was made.
Time stamp	Zoned(6,0)	The system time that the entry was made.
(Reserved area)	Char(95)	
Job name	Char(10)	The name of the job that caused the entry to occur.
User name	Char(10)	The user profile name associated with the job.
Job number	Zoned(6,0)	The job number.
Program name	Char(8)	The name of the MSF program that made the journal entry.
Function identifier	Char(1)	Function that was being performed when the entry was made. The possible values are: <ol style="list-style-type: none"> 1 MSF message created log entry 2 MSF message ended normally 3 MSF message reset by STRMSF command (STRMSF MSGOPT(*RESET)) 4 MSF message removed by STRMSF command (STRMSF MSGOPT(*CLEAR)) 5 MSF message acted on by address switcher
MSF message ID	Char(32)	The MSF message ID logged.
Length of entry data	Zoned(5,0)	The length of the logged data.
Logged data	Char(256)	The data logged by MSF when the function identifier is: <ol style="list-style-type: none"> 2 Data is three Zoned(5,0) numbers. The first number is the number of entries in the recipient list when the message was created. The second number is the number of entries in the recipient list when processing was completed for the message. The third number is the number of recipients that had a non-deliverable status when processing was completed for the message. 5 Data is two Zoned(5,0) numbers. The first number is the number of recipients that had their address switched by program QZMFSNPA. The second number is the total number of recipients that were in the recipient list of the MSF message processed by QZMFSNPA.

Figure B-9 is an example LG journal entry you might see when you use option 5 on the Display Journal Entries display to display the entire entry. The values of any QZMF journal entry field can be mapped to the entry data displayed.

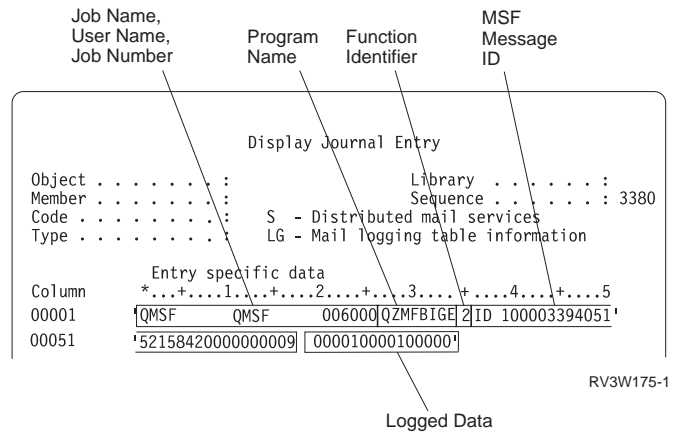


Figure B-9. Fields of the Journal Entries - Example

Format for MSF Message Errors (ER)

The entry for MSF message errors is mapped by the database file record, QAZMFER, that represents the error information entered. This record is defined by the physical file QAZMFER, which shipped in QSYS. The ER type journal entry contains the following information:

Table B-2. Type ER Journal Entries for MSF Message Errors

Field	Format	Description
Entry length	Zoned(5,0)	Total length of the journal entry, including the entry length field.
Sequence number	Zoned(10,0)	Applied to each journal entry. Initially set to 1 for each new or restored journal. Reset when a new receiver is attached.
Journal code	Char(1)	Always S for MSF entries.
Entry type	Char(2)	Always ER for MSF message error entries.
Date stamp	Char(6)	The system date that the entry was made.
Time stamp	Zoned(6,0)	The system time that the entry was made.
(Reserved area)	Char(95)	
Job name	Char(10)	The name of the job that caused the entry to occur.
User name	Char(10)	The user profile name associated with the job.
Job number	Zoned(6,0)	The job number.
Program name	Char(8)	The name of the MSF program that made the journal entry.
Error ID	Char(1)	The MSF error ID. The possible values are: 2 MSF message ended by an exit program 3 QMSF job ended by an exit program 4 An exit program returned data that was not valid 5 An exit program failed
MSF message ID	Char(32)	The MSF message ID logged.
Data length	Char(5)	The length of the logged data.
Logged data	Char(256)	The data logged by MSF when the error ID is: 2 Exit program name char(10) and library char(10) 3 Exit program name char(10) and library char(10) 4 Exit program name char(10) and library char(10) 5 Exit program name char(10) and library char(10)

Figure B-9 on page B-6 shows an example of how the fields defined for QZMF journal entries are mapped to the displayed data.

Format for MSF System Level Events Table (SY)

entries to be made in the MSF journal. The SY-type journal entry contains the following information:

The entry for system level events table change is shown by the database file record, QAZMFYSY, which represents MSF system name table entries. This record is defined by the physical file QAZMFYSY, which is shipped in the QSYS library. Various mail server framework functions cause

Table B-3. Type SY Journal Entries for the MSF System Level Events Table Changes

Field	Format	Description
Entry length	Zoned(5,0)	Total length of the journal entry, including the entry length field.
Sequence number	Zoned(10,0)	Applied to each journal entry. Initially set to 1 for each new or restored journal. Reset when a new receiver is attached.
Journal code	Char(1)	Always S for MSF entries.
Entry type	Char(2)	Always SY for MSF system level events entries.
Date stamp	Char(6)	The system date that the entry was made.
Time stamp	Zoned(6,0)	The system time that the entry was made.
(Reserved area)	Char(95)	
Job name	Char(10)	The name of the job that caused the entry to occur.
User name	Char(10)	The user-profile name associated with the job.
Job number	Zoned(6,0)	The job number.
Program name	Char(8)	The name of the MSF program that made the journal entry.
Function identifier	Char(1)	Function that was being performed when the entry was made. The possible values are: <ul style="list-style-type: none"> 1 STRMSF command started (QMSF jobs) 2 Internal tables initialized (part of STRMSF command function) 3 Internal queues initialized (part of STRMSF command function) 4 Space pool index created and initialized 5 ENDMSF command started (ended QMSF jobs) 6 Damaged or destroyed internal space found 7 Message destroyed by using DATAAREA QZMFKQ value 8 Abnormal IPL destroyed in internal MSF space index 9 MSF clean up functions (part of Reclaim Storage (RCLSTG) command) started A MSF space clean up function started B MSF clean up functions completed C MSF reclaim storage function ended
Data length	Zoned(5,0)	The length of the logged data.
Logged data	Char(256)	The data logged by MSF when the function identifier is: <ul style="list-style-type: none"> 6 32 character MSF message ID 7 32 character MSF message ID followed by total number of internal entries destroyed

Figure B-9 on page B-6 shows an example of how the fields defined for QZMF journal entries are mapped to the displayed data.

Format for MSF Configuration Changes (CF)

The entry for MSF configuration change is mapped by the database file record, ZMFXCFFT, which represents the change made. This record is defined by physical file QAZMFCF, which is shipped in the QSYS library. The CF-type journal entry contains the following information:

Table B-4. Type CF Journal Entries for MSF Configuration Changes

Field	Format	Description
Entry length	Zoned(5,0)	Total length of the journal entry, including the entry length field.
Sequence number	Zoned(10,0)	Applied to each journal entry. Initially set to 1 for each new or restored journal. Reset when a new receiver is attached.
Journal code	Char(1)	Always S for MSF entries.
Entry type	Char(2)	Always CF for MSF configuration change entries.
Date stamp	Char(6)	The system date that the entry was made.
Time stamp	Zoned(6,0)	The system time that the entry was made.
(Reserved area)	Char(95)	
Job name	Char(10)	The name of the job that caused the entry to occur.
User name	Char(10)	The user profile name associated with the job.
Job number	Zoned(6,0)	The job number.
Program name	Char(8)	The name of the MSF program that made the journal entry.
Function identifier	Char(1)	Function that was being performed when the entry was made. The possible values are: <ul style="list-style-type: none"> 1 MSF data type configured added to configuration database by QZMFCOPN program. MSF type tables initialized with shipped type definitions. 2 Reserved. 3 Added configuration to the configuration database. New MSF data type defined by using the QzmfAddMailCfg API. 4 Configuration removed from the configuration database. MSF data type removed by using the QzmfRmvMailCfg API. 5 Exit program removed from MSF exit point. 6 Exit program added to MSF exit point, except QIBM_QZMFMSF_VLD_TYP and QIBM_QZMFMSF_TRK_CHG. 7 Exit program added to QIBM_QZMFMSF_VLD_TYP or QIBM_QZMFMSF_TRK_CHG. A Install program started. B Install program ended. C Type not deleted during install. D Type not added during install. E Exit point program not deleted during install. F Exit point program not added during install.
Data length	Zoned(5,0)	The length of the logged data.
Logged data	Char(256)	The data logged by MSF when the function identifier is: <ul style="list-style-type: none"> 1 The record for the MSF data type that was added. 3 The record for the MSF data type that was added. 4 The record for the MSF data type that was removed. 5 The information about the MSF exit point program that was removed during install. 6 The information about the MSF exit point program that was added during install. 7 The information about the MSF exit point program that was added during install. C The record for the MSF data type that the install program failed to delete. D The record for the MSF data type that the install program failed to add. E The information about the MSF exit point program that the install program failed to delete. F The information about the MSF exit point program that the install program failed to add.

Figure B-9 on page B-6 shows an example of how the fields defined for QZMF journal entries are mapped to the displayed data.

Figure B-10 shows an example of how the A function identifier field is mapped to the displayed data.

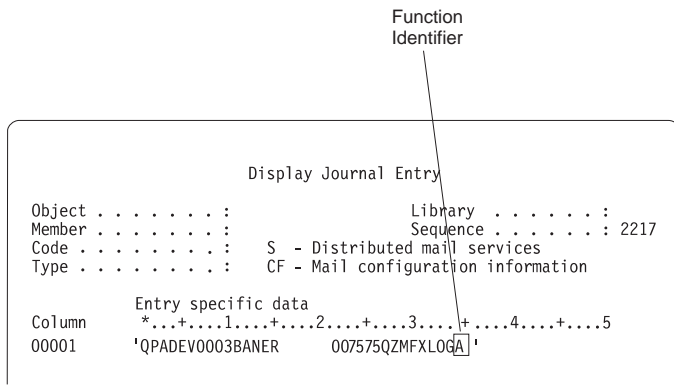


Figure B-10. Function Identifier A - Example

Figure B-11 shows an example of how an B function identifier field is mapped to the displayed data.

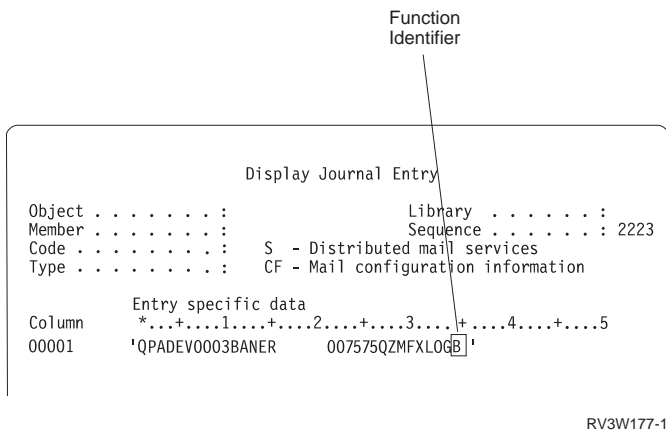


Figure B-11. Function Identifier B - Example

Figure B-12 shows an example of how an D function identifier field is mapped to the displayed data. The C and D function identifier entries look identical.

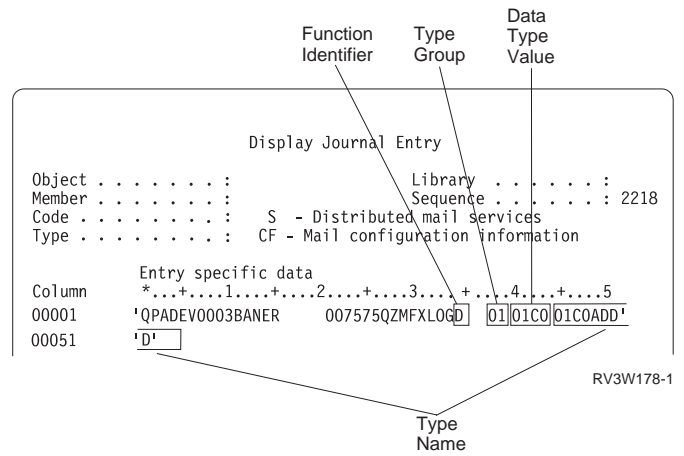


Figure B-12. Function Identifier D - Example

Figure B-13 shows an example of how an F function identifier field is mapped to the displayed data. The E and F function identifier entries look identical.

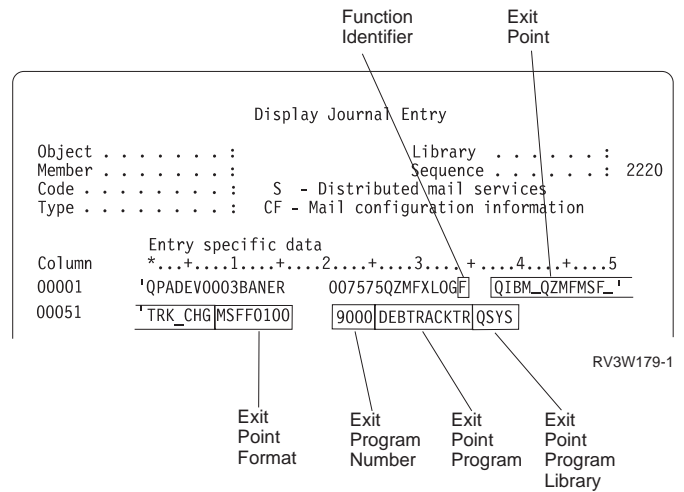


Figure B-13. Function Identifier F - Example

Appendix C. Preregistered Exit Point Programs

The exit point programs shipped with MSF are preregistered when you install your AS/400 system. These preregistered programs are shown in Figure C-1.

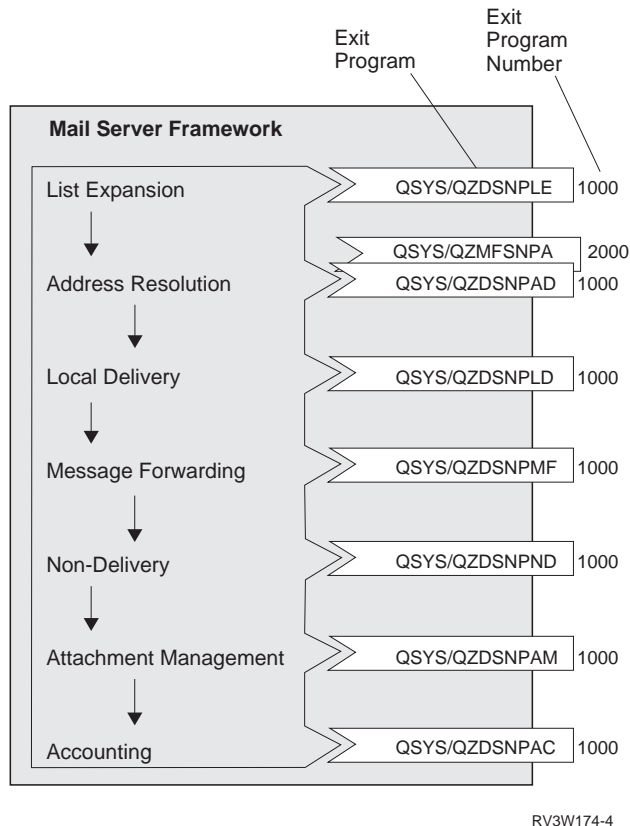


Figure C-1. Exit Point Programs Shipped with MSF

On the Work with Registration Information display, type 8 in the *Opt* Field next to the address resolution exit point, QIBM_QZMFMSF_ADR_RSL. The Work with Exit Programs display containing information about the address resolution exit point is shown, as in Figure C-2. The two programs, QZDSNPAD and QZMFSNPA, are the address resolution handler and the switcher programs.

```

Work with Exit Programs
Exit point:  QIBM_QZMFMSF_ADR_RSL      Format:  MSFF0100
Type options, press Enter.
1=Add  4=Remove  5=Display  10=Replace

Opt      Exit Program Number  Exit Program      Library
---      -
0
1000    QZDSNPAD             QSYS
2000    QZMFSNPA             QSYS

Command
====>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel
    
```

Figure C-2. Address Resolution Exit Point Programs

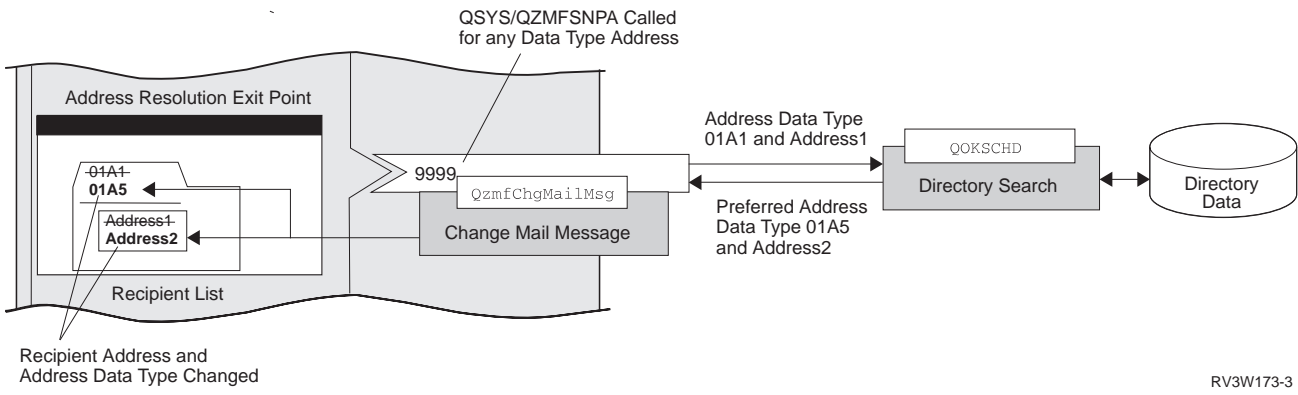
Exit Program QZMFSNPA

For those recipients of a MSF message who do not have a MSF data type assigned by previous address resolution exit point programs, a preregistered default exit point program is called. This preregistered address resolution exit point program, QZMFSNPA, changes the recipient list addresses to their preferred addresses and address types, as specified in the system distribution directory. The exit point program uses the directory search API (QOKSCHD) to determine the preferred addresses and address types of the message recipients. An example is shown in Figure C-3 on page C-2.

Note

For any exit point program, including QZMFSNPA, to use the directory search API (QOKSCHD), the directory must allow searches. The Change System Directory Attributes (CHGSYSDIRA) command can be used to specify that directory searches are allowed by changing the ALWSCH parameter to *YES. If directory searches are not allowed, QZMFSNPA issues error message CPFAF90, which indicates the QMSF job will end.

See the *SNA Distribution Services* book for more information about preferred addresses.



RV3W173-3

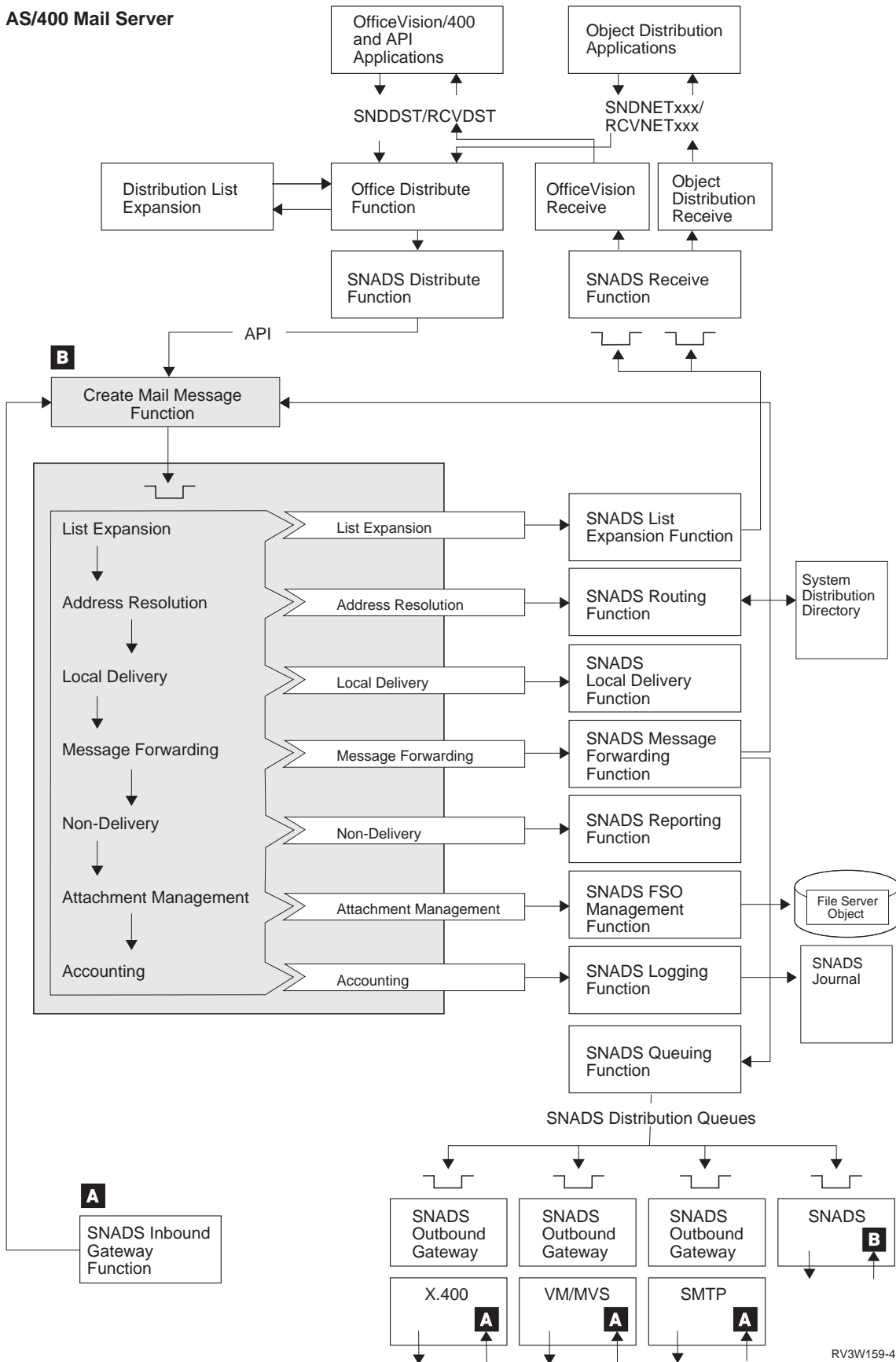
Figure C-3. Example of Preferred Address Resolution

Appendix D. How Mail Server Framework Works with SNADS, Object Distribution, and OfficeVision

This appendix describes how SNA distribution services (SNADS), as shipped by IBM, uses the mail server framework (MSF) to route and distribute messages asynchronously. The function of SNADS has not changed because of the introduction of the MSF. The SNADS routing function

has been organized differently. Applications that use SNADS, such as OfficeVision and Object Distribution, are not affected by this change to SNADS. Figure D-1 on page D-2 shows the current organization of SNADS.

AS/400 Mail Server



RV3W159-4

Figure D-1. SNADS Overview

The routing function of SNADS occurs within the exit point programs that are configured for the mail server framework exit points. Only seven of the ten available exit points are used by SNADS.

- List expansion
- Address resolution
- Local delivery
- Message forwarding
- Non-delivery
- Attachment management
- Accounting

List Expansion

The SNADS exit program for list expansion expands a distribution list into a recipient list containing SNADS recipients. This exit program accepts the following address type:

Type	Description
ATDDLST	Distribution list

Address Resolution

In the exit point program for address resolution, SNADS accesses the system distribution directory, the SNADS routing table, the SNADS distribution queues table, and the SNADS secondary system name table to determine the correct route of a message recipient. This exit program accepts the following address types:

Type	Description
ATDGNDEN	Address/User ID
ATRGEDGE	System name/Address/User ID (fully qualified recipient name)
ATRGNREN	Group name/System name
ATDDLST	Distribution list

Information that is stored in the system distribution directory (for example, preferred address and mail service level) can be used in determining the message type of a recipient. Preferred address is generally used to determine the address type of a remote recipient (for example, TCP/IP or X.400). Mail service level is used to decide how the message will be stored for a local recipient. If the address is stored in the system distribution directory, the directory search API (QOKSCHD) can be used to query information needed by the user.

If a recipient is local, the SNADS address resolution exit program checks the system directory to see what the mail service level is for that recipient. If the mail service level is user index, this exit program continues to process the recipient. If the mail service level is system message store or other mail service, this exit program does not route the recipient, and lets another address resolution exit program process it.

If a recipient is remote, the SNADS address resolution exit program checks the system directory to see what address

preference is stored for the recipient. If the preference is either the ATDGNDEN or the ATRGEDGE address type, then this exit program continues to route the recipient. If the preference is any other address type, this exit program does not route the recipient, and lets another address resolution exit program process it.

After determining the correct route for a recipient, the SNADS address resolution exit program assigns a message type and status (local or remote) for that recipient. The SNADS message types include:

Type	Description
MTDIA	OfficeVision/400 type
MTOBJDST	Object Distribution type
MTDLS	Document Library Services type
MTDSNX	Distributed System Node Executive type

Local Delivery

The SNADS exit program for local delivery routes a message to the local Object Distribution or OV/400 applications for all local recipients. This exit program accepts the following message types:

Type	Description
MTDIA	OfficeVision/400 type
MTOBJDST	Object Distribution type
MTDLS	Document Library Services type
MTDSNX	Distributed System Node Executive type

Message Forwarding

The SNADS exit program for message forwarding routes a message to the appropriate SNADS distribution queue for all remote recipients. This exit program accepts the following message types:

Type	Description
MTDIA	OfficeVision/400 type
MTOBJDST	Object Distribution type
MTDLS	Document Library Services type
MTDSNX	Distributed System Node Executive type

Non-Delivery

The SNADS exit program for non-delivery creates a status message for those recipients who had routing or delivery errors. This exit program then uses the Create Mail Message API to put the new status message into the mail server framework to be routed. This exit program accepts the following message types:

Type	Description
MTDIA	OfficeVision/400 type
MTOBJDST	Object Distribution type
MTDLS	Document Library Services type
MTDSNX	Distributed System Node Executive type

Attachment Management

The SNADS exit program for attachment management manages the SNADS file server object attached to the message, if there is one. File server objects include files, spool files, or network jobs sent by object distribution, or documents or notes sent by OV/400. This exit program accepts the following message types:

Type	Description
MTDIA	OfficeVision/400 type
MTOBJDST	Object Distribution type
MTDLS	Document Library Services type
MTDSNX	Distributed System Node Executive type

Accounting

The SNADS exit program for accounting makes entries in the QSNADS journal for any messages that are processed. The function types or entry types that can be journaled by this exit program include the following:

*RTR/*NRM
*RTR/*ERR

The Display Distribution Log (DSPDSTLOG) command can be used to view the SNADS journal entries. See the *SNA Distribution Services* book for more information on SNADS journaling.

Bibliography

The IBM publications listed here provide additional information about topics described or referred to in this book. These books are listed with their full title, order number, and version and release number.

- *AnyMail/400 Mail Server Framework Developer Guide*, GG24-4449
provides the programmer technical details to understand how the AnyMail/400 mail server framework (MSF) works. It provides information about how to write programs that either create MSF messages or how to act on MSF messages as MSF exit point programs.

This book is written to assist AS/400 programmers who use MSF to write mail messaging applications. Its scope includes a description of MSF, designing MSF applications, configuring MSF, MSF operations, and MSF problem determination.
- *CL Reference*, SC41-5722
provides the application programmer with a description of the AS/400 control language (CL) and its commands. Each command description includes a syntax diagram, parameters, default values, keywords, and an example.
- *Communications Configuration*, SC41-5401
contains general configuration information including detailed descriptions of network server, network interface, line, controller, device, mode, class-of-service, and NetBIOS descriptions, and configuration lists and connection lists.
- *Alerts Support*, SC41-5413
provides the system operator, programmer, or system administrator with information for configuring and using AS/400 Alerts support. It discusses how to allow end-user applications to create alerts, control the creating and sending of alert messages for problem management, and perform central site problem analysis for the AS/400 systems in the network.
- *Publications Reference*, SC41-5003
identifies and describes the printed and online information in the AS/400 library, and also lists other publications about the AS/400 system. Includes cross-reference information between the current library and the previous version library.
- *SNA Distribution Services*, SC41-5410
provides the system operator or system administrator with information about configuring a network for Systems Network Architecture distribution services (SNADS) and the Virtual Machine/Multiple Virtual Storage (VM/MVS) bridge. In addition, object distribution functions and document library and system distribution directory services are discussed.
- *System API Reference*, SC41-5801
provides information for those customers or systems houses that wish to:
 - Write their own communications protocol on the AS/400 system to connect to systems in ways not currently possible with IBM communications support, or
 - Connect programmable work stations (PWSs) through a specialized Virtual Terminal Manager interface.
- *TCP/IP Configuration and Reference*, SC41-5420
provides information for configuring and using AS/400 TCP/IP support. The applications included are Network Status (NETSTAT), Packet InterNet Groper (PING), TELNET, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), line printer requester (LPR), and line printer daemon (LPD). The TCP and UDP Pascal application program interface (API) is also discussed.
- *Work Management*, SC41-5306
provides information about creating and using subsystems.

Index

Special Characters

*TYPE1 B-1

A

accounting exit point 4-9

Add Mail Server Framework Configuration
(QzmfAddMailCfg) API A-1

adding types 3-2

address resolution exit point 4-4

AnyMail/400 1-1

API (application program interface)

definition 1-1

mail server framework A-1

attachment conversion exit point 4-6

attachment management exit point 4-9

attachment reference

list 5-3

B

benefits vii

bibliography H-1

C

CF journal entry B-1

Change Mail Message (QzmfChgMailMsg) API A-2

code S B-1

command, CL

End Mail Server Framework (ENDMSF) 2-1

ENDMSF (End Mail Server Framework) 2-1

Start Mail Server Framework (STRMSF) 2-1

STRMSF (Start Mail Server Framework) 2-1

Complete Creation Sequence (QzmfCrtCmpMailMsg)

API A-2

configuration

considerations 3-1

configuration APIs A-1

considerations

configuration 3-2

description 2-3

operations 2-1

Create Mail Message (QzmfCrtMailMsg) API A-1

D

data type

address 5-4

attachment reference 5-4

definition 5-4

envelope 5-4

message 5-4

data type (continued)

rules 5-4

description considerations 2-3

displaying

journal B-1

E

End Mail Server Framework (ENDMSF) command 2-1

ending MSF jobs 2-2

ENDMSF (End Mail Server Framework) command 2-1

envelope

definition 4-6

generic 5-3

list 5-3

envelope processing exit point 4-6

ER journal entry B-1

error messages 2-3

error recovery 2-3

example

A function identifier B-11

B function identifier B-11

D function identifier B-11

exit point program 3-1

F function identifier B-11

journal entry fields B-6

registered exit point program 3-1

registering a program 3-1

exit point

See also System API Reference

accounting 4-9

address resolution 4-4

attachment conversion 4-6

attachment management 4-9

definition 1-1, 4-1

envelope processing 4-6

example 4-1

list expansion 4-4

local delivery 4-8

message forwarding 4-8

non-delivery 4-9

security and authority 4-7

exit point program

addressing group 4-3

definition 4-1

delivery group 4-3

management group 4-3

multiple 4-2

overview 4-2

pre-delivery processing group 4-3

preregistered C-1

QZMFSNPA C-1

registered 3-1

exit point program *(continued)*

- snap-in call A-2
- snap-in definition 1-1
- track mail message changes A-2
- validate data field A-2

exit program

- definition 1-1

F

formatting

- QZMF journal B-1

framework types

- adding 3-2

G

generic envelope type 5-3

J

jobs, MSF

- managing 2-2

journal

- displaying B-1
- entries
 - displaying QZMF B-1
 - type CF B-4, B-9
 - type ER B-3, B-7
 - type LG B-3, B-5
 - type SY B-3, B-8

QZMF

- configuration changes (CF) B-9
- displaying B-1
- error entry (ER) B-7
- formatting B-1
- logging (LG) B-5
- support B-1
- system level events table (SY) B-8

receiver

- full B-1
- new B-9

L

LG journal entry B-1

list

- attachment reference 5-3
- envelope 5-3
- original recipient 5-2
- originator 5-2
- recipient 5-2
- report-on 5-3
- report-to 5-3

list expansion exit point 4-4

List Mail Server Framework Configuration (QzmfLstMailCfg) API A-1

local delivery exit point 4-8

M

mail server framework (MSF)

- adding types 3-2
- API A-1
- benefits vii
- configuration considerations 3-2
- definition 1-1
- ending 2-1
- introduction 1-1
- managing jobs 2-2
- starting 2-1
- using 2-1

mail server framework API A-1

message

- no log entries B-3

message APIs A-1

message forwarding exit point 4-8

message identifier 5-3

message lists 5-2

MSF (mail server framework)

- adding types 3-2
- API A-1
- benefits vii
- configuration considerations 3-2
- definition 1-1
- ending 2-1
- introduction 1-1
- managing jobs 2-2
- starting 2-1
- using 2-1

MSF API A-1

N

non-delivery exit point 4-9

O

object distribution D-1

OfficeVision D-1

operations considerations 2-1

original recipient

- list 5-2

originator

- list 5-2

outfile format (OUTFILFMT) parameter B-1

OUTFILFMT (outfile format) parameter B-1

P

preregistered exit point program C-1

program

- registered 3-1

programming example

See AnyMail/400 Mail Server Framework Developer Guide publications, list of H-1

Q

QSYSWRK

jobs 2-2
subsystem 2-2

Query Mail Message Identifier (QzmfQryMailMsgId)

API A-2

QZMF journal B-1

QzmfAddMailCfg (Add Mail Server Framework Configuration) API A-1

QzmfChgMailMsg (Change Mail Message) API A-2

QzmfCrtCmpMailMsg (Complete Creation Sequence)

API A-2

QzmfCrtMailMsg (Create Mail Message) API A-1

QzmfLstMailCfg (List Mail Server Framework Configuration) API A-1

QzmfQryMailMsgId (Query Mail Message Identifier)

API A-2

QzmfRmvMailCfg (Remove Mail Server Framework Configuration) API A-1

QzmfRmvRsvMailMsgId (Remove Reserved Mail Message Identifier) API A-2

QzmfRsvMailMsgId (Reserve Mail Message Identifier)

API A-2

QzmfRtvMailMsg (Retrieve Mail Message) API A-2

QZMFSNPA exit point program C-1

R

recipient

list 5-2

registering exit point programs 3-1

related printed information H-1

Remove Mail Server Framework Configuration (QzmfRmvMailCfg) API A-1

Remove Reserved Mail Message Identifier

(QzmfRmvRsvMailMsgId) API A-2

report-on

list 5-3

report-to

list 5-3

Reserve Mail Message Identifier (QzmfRsvMailMsgId)

API A-2

Retrieve Mail Message (QzmfRtvMailMsg) API A-2

routing function

See SNA Distribution Services

routing table

See SNA Distribution Services

rules

data type definitions 5-4

S

S code B-1

security and authority exit point 4-7

SNADS

graphic overview D-1
routing function D-3
using 7 exit points D-3
using mail server framework D-1

snap-in

definition 1-1, 4-1
example 4-1

Snap-In Call Exit Point Program A-2

Start Mail Server Framework (STRMSF) command 2-1

starting MSF jobs 2-2

STRMSF (Start Mail Server Framework) command 2-1

SY journal entry B-1

T

Track Mail Message Changes Exit Point Program A-2

trouble shooting 2-3

type

journal entries
CF B-9
ER B-7
LG B-5
SY B-8
MSF data 3-2, 5-4

V

Validate Data Field Exit Point Program A-2

Reader Comments—We'd Like to Hear from You!

AS/400 Advanced Series
 AnyMail/400 Mail Server
 Framework Support
 Version 4

Publication No. SC41-5411-00

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

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

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
T H A N K Y O U !				

Please tell us how we can improve this manual:

May we contact you to discuss your responses? Yes No
 Phone: (____) _____ Fax: (____) _____ Internet: _____

To return this form:

- Mail it
- Fax it
 - United States and Canada: **800+937-3430**
 - Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

 Name

 Address

 Company or Organization

 Phone No.



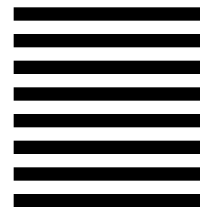
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 542 IDCLERK
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-9986



Fold and Tape

Please do not staple

Fold and Tape



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

SC41-5411-00



Spine information:



AS/400 Advanced Series

AnyMail/400 Mail Server
Framework Support

Version 4