

GXS EDI Services



Expedite Notification Manager for AS/400 Programming Guide

Version 1 Release 1

Second Edition (November 2005)

This book is current as of its edition date. New editions will reflect changes in procedures or technical details.

© Copyright GXS, Inc. 1998, 2005. All rights reserved.

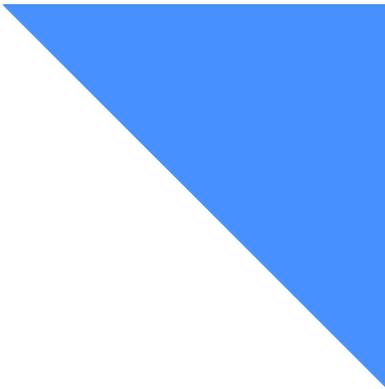
Government Users Restricted Rights - Use, duplication, or disclosure restricted.



Contents

To the reader	v
Who should read this book	v
Command syntax conventions	v
What this book contains	vi
Related books	vi
Chapter 1. Introducing Expedite Notification Manager	1
System diagram	2
Chapter 2. Installing Expedite Notification Manager	3
Ordering the logical unit (LU) name	3
Installing the Expedite Notification Manager Program	3
Chapter 3. Expedite Notification Manager files	5
How Expedite Notification Manager uses the notification log	5
Notification log description	5
Notification log (ENMNTFY)	5
Optional files	7
Profile command file (ENMIPRO)	7
Profile response file (ENMOPRO)	7
Error message file (ENMERROR)	8
Extended error text file (ENMTEXT)	8
Logic trace file (ENMTRACE)	8
Link trace file (ENMLINK)	8
Run-time information (ENMRUN)	9
Chapter 4. Expedite Notification Manager profile commands	11
Expedite Notification Manager profile (ENMIPRO)	11
Expedite Notification Manager profile commands	11
EXECUTE command	12
TRACE command	13
Profile response records	14

PROFILERC record	14
RETURN record	14
Chapter 5. Receiving Notification Data	17
Notification log information	18
EVENT record	18
RETURN record	22
Notification log examples	23
Chapter 6. Application Examples	25
Application example 1	25
Application example 2	26
Appendix A. Expedite Notification Manager logs and traces	27
Expedite Notification Manager logs	28
Notification log	28
Run log	28
Expedite Notification Manager traces	29
Link trace - TRACE LINK(Y)	29
Logic trace - TRACE LOGIC(Y)	29
Appendix B. Messages and codes	33
Glossary	39
Index	43



To the reader

This book describes Expedite Notification Manager and its functions.

Who should read this book

This book is intended for users who have a basic understanding of Expedite and Information Exchange products, and the AS/400 environment.

Command syntax conventions

To use Expedite Notification Manager, you need to understand the basic command syntax and how to use the profile commands.

The following is an example of Expedite Notification Manager command syntax:

```
command parameter(value) parameter(value) ...parameter(value);
```

Descriptions of the syntax elements are:

Syntax Element	Description
#	Defines or delimits a comment line. You can type any information you like after a #, and Expedite Notification Manager ignores the characters that follow it on the same line. If you include a # in a parameter value, Expedite Notification Manager knows the # is part of a command and does not ignore the parameter value of the characters that follow it.
command	Identifies the Expedite Notification Manager command
parameter	Identifies a parameter of the associated command. <ul style="list-style-type: none">• All parameter values are shown in italics• Required parameter values are shown in boldface• Default parameter values are underlined.
value	Defines the value associated with the parameter.

Syntax Element	Description
...	In the previous example, the ellipsis (...) indicates that you can specify as many parameters as necessary. (It is not part of the syntax).
;	Ends the command

You can type Expedite Notification Manager commands and parameters in uppercase or lowercase letters. The commands and parameters can span several lines in the Expedite Notification Manager profile. However, the following limitations apply:

- Type the entire command name; for example, **trace**, on a single input line.
- Type the entire parameter name; for example, **link**, on a single input line.
- Ensure that each parameter is immediately followed by a left parenthesis. Do not use spaces between parameter names and values. For example, type **link(y)** rather than **link (y)**.
- End each command with a semicolon.

What this book contains

This book contains the following:

- Chapter 1, "Introducing Expedite Notification Manager," gives an overview of Expedite Notification Manager.
- Chapter 2, "Installing Expedite Notification Manager," provides general information about how Expedite Notification Manager is installed.
- Chapter 3, "Expedite Notification Manager files," provides a general explanation of how Expedite Notification Manager works, and describes the files needed.
- Chapter 4, "Expedite Notification Manager profile commands," explains how to create profiles, and describes the profile commands and profile response records.
- Chapter 5, "Receiving notification data," describes the format of information written to the Expedite Notification Manager notification log, and provides examples.
- Chapter 6, "Application examples," describes how applications are written using the information supplied in the event notification.
- Appendix A, "Expedite Notification Manager logs and traces," provides examples of Expedite Notification Manager logs and traces.
- Appendix B, "Messages and codes," provides the Expedite Notification Manager return codes with explanations and user actions.

Related books

AS/400 Communications: User's Guide, SC21-9601

Communications: Operating System/400 Communications Configuration Reference, SC41-0001

Expedite Base/400 Programming Guide, GC34-2254

Expedite 400 Programming Guide, GC34-3002

Using Information Exchange Administration Services, GC34-2221

Introducing Expedite Notification Manager

Expedite Notification Manager works with Information Exchange to provide event-driven notifications. An event-driven notification is when Information Exchange notifies Expedite Notification Manager that an event has occurred on Information Exchange. Based on your event profile in Information Exchange Administration Services, Information Exchange notifies Expedite Notification Manager using a notification that contains data about the event that occurred.

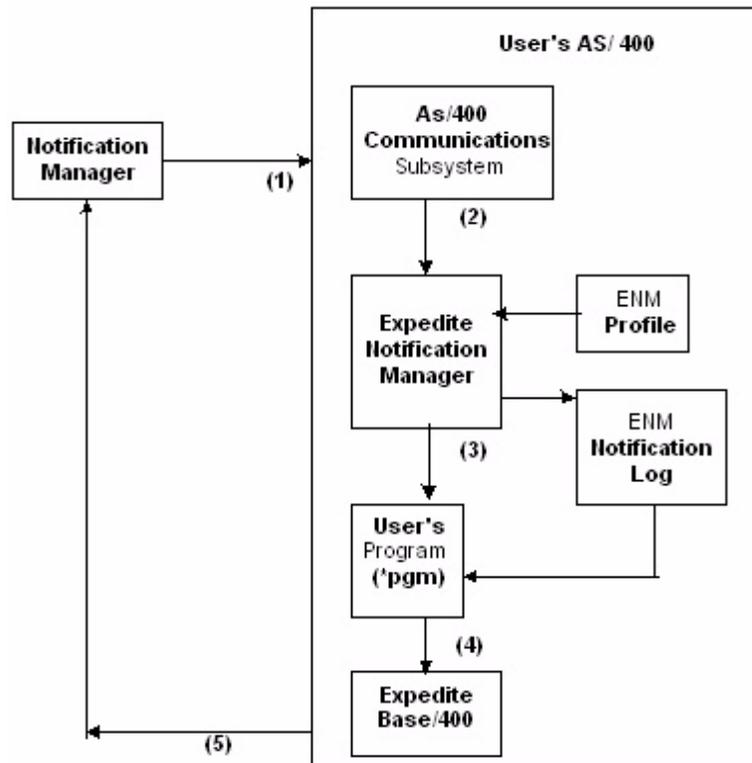
There are three types of events:

- Message-arrival** A message-arrival event occurs when a message arrives at your Information Exchange mailbox. You can have notifications sent for every message, or you can create message-arrival definitions using Information Exchange Administration Services to limit the number of notifications sent to Expedite Notification Manager.
- Scheduled** A scheduled event occurs based on a schedule defined in Information Exchange Administration Services. You can specify the date, time, and frequency of the schedule for sending notifications.
- User-initiated** A user-initiated event occurs when your Information Exchange administrator requests a notification to be sent.

When a notification is received, Expedite Notification Manager logs the notification in a file. Expedite Notification Manager can also start an application in response to the notification. This application can be Expedite 400 Base, Expedite Base/400, or a user-written application.

System diagram

The following is the system diagram for Expedite Notification Manager. This pictorial overview represents a complete transaction beginning with a message-arrival, schedule-based, or user-initiated event at Information Exchange and ending with the invocation of an application in your system by Expedite Notification Manager. In this case, the user application invokes Expedite Base/400 to retrieve a mail item from Information Exchange. This process flow is repeated for each event triggered in Information Exchange.



1. Information Exchange Notification Manager allocates a conversation with Expedite Notification Manager via the AS/400 Communications Subsystem when an event occurs.
2. AS/400 communications subsystem processes the program start request.
3. Expedite Notification Manager receives the notification and, based on Expedite Notification Manager profile commands, writes the information to the Expedite Notification Manager notification log. If the Userappl field in the notification contained a value and the profile command file contained EXECUTE USERAPPL(1), Expedite Notification Manager calls the program.
4. Optionally, your program can then process the Expedite Notification Manager notification log to produce input to Expedite, check the status of the last Expedite Base/400 run, and submit Expedite Base/400.
5. Expedite Base/400 receives the data from your Information Exchange mailbox.

Installing Expedite Notification Manager

This chapter contains an overview of the installation procedure that must be performed by your system administrator.

Ordering the logical unit (LU) name

Before you can use Expedite Notification Manager, you must have a leased-line attached system, and you must order the LU name that Expedite Notification Manager needs to communicate with Information Exchange LU62 Event Services. The LU ordered is referred to as *luname* in the following instructions.

Your GXS sales representative can help you with your order.

Installing the Expedite Notification Manager Program

The distribution media you received from GXS contains the programs and procedures needed to operate Expedite Notification Manager.

Detailed installation steps are available on the GXS product website:

<https://www.gxsolc.com/public/EDI/us/support/Products/expedite/install.html>

Expedite Notification Manager files

This chapter provides information about the Expedite Notification Manager files. It provides a general explanation of how Expedite Notification Manager uses the files and their attributes.

How Expedite Notification Manager uses the notification log

Expedite Notification Manager is an event-oriented application. Expedite Notification Manager runs in response to an event occurring in Information Exchange. Information Exchange notifies the user that an event has occurred. Expedite Notification Manager is started based on a notification received from Information Exchange.

Expedite Notification Manager receives the event notification and writes it to the notification log ENMNTFY.

Because Information Exchange events can occur in rapid succession, care should be taken when developing user applications that process the notification log. A sample program is provided that causes events to be logged into event-unique ENMNTFY members.

Notification log description

The notification log that all Expedite Notification Manager users are required to have is described here. You might need to use other files as well, depending on your applications. All Expedite Notification Manager files can be created using the following command:

```
CRTPF ENM400/filename RCDLEN (record length)
```

The filename and record-length values are provided in this chapter. All files described in this chapter are installed with the program product.

Notification log (ENMNTFY)

The name of the notification log is ENMNTFY. This file is used by Expedite Notification Manager to write the received event notification. It also contains a return code from Expedite Notification Manager processing of the event notification.



NOTE: Expedite Notification Manager appends the data to this file. Periodically clear this file; otherwise, it can become quite large.

ENMNTFY attributes

ENMNTFY has the following attributes:

This attribute	Has these characteristics
Space	ENMNTFY has no minimum space requirement. It must be large enough to hold unprocessed notifications.
Record length	ENMNTFY has a record length of 80 bytes.

ENMNTFY example

An ENMNTFY example log is shown in Chapter 4, "Receiving notification data."

Optional files

The following are optional Expedite Notification Manager files.

Profile command file (ENMIPRO)

The name of the profile command file is ENMIPRO. Expedite Notification Manager uses the information contained in ENMIPRO to determine whether or not to start a user-supplied application or to produce trace, log, or link output.

Defaults exist for all command parameters contained in the profile command file. However, if you want to override the default parameter values, the overrides must be provided in this file.

ENMIPRO attributes

ENMIPRO has the following attributes:

This attribute	Has these characteristics
Space	ENMIPRO has no minimum space requirement. It must be large enough to hold all the profile commands.
Record length	ENMIPRO has a record length of 80 bytes.

ENMIPRO example

The following is an example of an ENMIPRO file:

```
execute userappl(1);
trace log(y);
```

Profile response file (ENMOPRO)

The name of the profile response file is ENMOPRO. When Expedite Notification Manager reads the profile command file, it echoes the profile commands to ENMOPRO, along with the associated return codes. The return record in ENMOPRO contains the return codes for each command. The PROFILERC record contains the return code for the processing of the entire ENMIPRO file.

ENMOPRO attributes

ENMOPRO has the following attributes:

This attribute	Has these characteristics
Space	ENMOPRO must be large enough to hold the echoed ENMIPRO commands and their associated responses
Record length	ENMOPRO has a record length of 80 bytes.

ENMOPRO example

The following is the ENMOPRO file produced by the profile command file:

```
execute userappl(1);

RETURN(ööööö);

trace log(y);

RETURN(ööööö);

PROFILERC(ööööö);
```

Error message file (ENMERROR)

Expedite Notification Manager can provide error descriptions in the RETURN and PROFILERC response records. These error descriptions are contained in the ENMERROR file.

Extended error text file (ENMTEXT)

Expedite Notification Manager can provide extended error descriptions in the RETURN and PROFILE response records. These extended error descriptions are contained in the ENMTEXT file.

Logic trace file (ENMTRACE)

The name for the logic trace file is ENMTRACE. It contains Expedite Notification Manager module execution information. You need to request a trace only when working with GXS on a problem.

ENMTRACE attributes

ENMTRACE has the following attributes:

This attribute	Has these characteristics
Space	ENMTRACE produces approximately 100 80-byte records.
Record length	ENMTRACE has a record length of 80 bytes.

Link trace file (ENMLINK)

The name for the link trace file is ENMLINK. It contains communication information produced by the link trace. You need to request a link trace only when working with GXS on a problem.

ENMLINK attributes

ENMLINK has the following attributes:

This attribute	Has these characteristics
Space	ENMLINK produces approximately 100 121-byte records.
Record length	ENMLINK has a record length of 121 bytes.

Run-time information (ENMRUN)

The name for the run-time information file is ENMRUN. You can request run-time information using the trace profile command.



NOTE: Expedite Notification Manager appends data to this file. Periodically clear this file; otherwise, it can become quite large.

ENMRUN attributes

ENMRUN has the following attributes:

This attribute	Has these characteristics
Space	ENMRUN produces 15 80-byte records per execution of Expedite Notification Manager.
Record length	ENMRUN has a record length of 80 bytes.

ENMRUN example

An example of an ENMRUN file is shown in Appendix A.

Expedite Notification Manager profile commands

You can use the Expedite Notification Manager profile to allow various Expedite Notification Manager options such as tracing, or to provide information to Expedite Notification Manager such as execution options. This chapter discusses the Expedite Notification Manager profile and describes the profile commands and response records. The Expedite Notification Manager profile is optional. If the profile is empty, the default values will be used.

Expedite Notification Manager profile (ENMIPRO)

To use the Expedite Notification Manager profile, enter the profile commands in the profile command file. Expedite Notification Manager echoes these commands along with the response records and their associated return codes to the profile response file (ENMOPRO). You can review ENMOPRO to verify successful completion of the profile commands.

Expedite Notification Manager profile commands

The following commands can be used in your Expedite Notification Manager profile.

EXECUTE

Use this command to specify execution options for Expedite Notification Manager.

TRACE

Use this command to specify the information that Expedite Notification Manager records into the trace files.

EXECUTE command

The EXECUTE command specifies how to process the notification.

The syntax of the EXECUTE command is:

```
execute  
  
userappl (0|1)  
appldata (0|1);
```

userappl

Indicates whether Expedite Notification Manager should try to start the program specified in the **User Application** field passed from the Information Exchange Administration Services profile.

- 0** Do not start the user application. This is the default.
- 1** Try to start the application specified in the **User Application** field.

The application program is submitted to the job queue specified in the default user job description.

appldata

Indicates whether Expedite Notification Manager should pass Information Exchange Administration Services profile application data values as parameters for the called **userappl**.

- 0** Do not pass the application data.
- 1** Pass the application data. This is the default.



NOTE: **appldata** is valid only if **userappl** is set to 1.

EXECUTE command example

```
execute userappl(1);
```

TRACE command

The TRACE command specifies what information is recorded in the trace files during a session. Possible traces are LINK, LOGIC, and LOG. All traces are used for problem determination. Trace information is written to the appropriate file.

The syntax of the TRACE command is:

```
trace  
  
link(n|y)  
logic(n|y)  
log(n|y);
```

link

Indicates whether or not Expedite Notification Manager should trace communication link protocol. Expedite Notification Manager writes the communication protocol data to the file ENMLINK.

- n** Do not write the communication link information to the file ENMLINK. This is the default.
- y** Write the communication link information to the file ENMLINK.

logic

Indicates whether the file ENMTRACE should contain the Expedite Notification Manager module execution information.

- n** Do not write the Expedite Notification Manager module execution information to ENMTRACE. This is the default.
- y** Write the Expedite Notification Manager module execution information to the file ENMTRACE.

log

Indicates if Expedite Notification Manager run-time information should be recorded.

- n** Do not write the Expedite Notification Manager run-time information to the file ENMRUN. This is the default.
- y** Write the Expedite Notification Manager run-time information to the file ENMRUN.

TRACE command example

```
trace link(y);
```

Profile response records

The profile response file ENMOPRO contains an echo of the profile commands and their response records.

The profile response records are:

- **PROFILERC**

This record indicates the completion of ENMIPRO.

- **RETURN**

This record indicates the completion of a command in ENMIPRO.

The following sections provide detailed information on each of these records.

PROFILERC record

The PROFILERC record is the last record in ENMOPRO. The PROFILERC record indicates the processing of the profile commands is complete. A zero value indicates that all the profile commands completed successfully.

The following example shows the format of the PROFILERC record:

```
PROFILERC(return code) ERRDESC(error description)
```

```
ERRTEXT(error text);
```



NOTE: Each ENMOPRO has only one PROFILERC record.

PROFILERC

Indicates whether Expedite Notification Manager processed the profile commands successfully. If the return code is zero, the commands completed successfully. If the return code is not zero, the program displays an error number along with ERRDESC and ERRTEXT records. This parameter contains 5 numeric characters.

ERRDESC

Provides a short description of an error. If the return code is zero, this parameter is not in ENMOPRO. This parameter contains 1 to 76 alphanumeric characters.

ERRTEXT

Provides a detailed description of an error and may suggest steps to correct the problem. There may be multiple error text records in the file. If the return code is zero, this parameter is not in ENMOPRO. This parameter contains 1 to 76 alphanumeric characters.

RETURN record

The RETURN record indicates the completion of a command in ENMIPRO. A zero value indicates that the command completed successfully.

The following example shows the format of the RETURN record:

```
RETURN(return) ERRDESC(error description)
```

```
ERRTEXT(error text);
```

RETURN

Indicates completion of an Expedite Notification Manager profile command. If the return code is zero, the command completed successfully. If the return code is not zero, the program displays an error number along with ERRDESC and ERRTEXT records. This parameter contains 5 numeric characters.

ERRDESC

Provides a short description of an error. If the return code is zero, this parameter is not in ENMOPRO. This parameter contains 1 to 76 alphanumeric characters.

ERRTEXT

Provides a detailed description of an error and may suggest steps to correct the problem. There may be multiple error text records in the file. If the return code is zero, this parameter is not in ENMOPRO. This parameter contains 1 to 76 alphanumeric characters.

Receiving Notification Data

Expedite Notification Manager places all notification data in the notification log. The Expedite Notification Manager notification log contains the data from notifications, such as event date, event time, event type, final return code data, and possible error text. Expedite Notification Manager only logs notifications received in Expedite Notification Manager format (as specified in the Information Exchange event profile). New notifications are appended to the end of this file; therefore, it can become very large and needs maintenance at regular intervals.

All the parameters shown here may not be included in the notification log, because Expedite Notification Manager does not write parameters with blank values. The parameters received by Expedite Notification Manager depend on the notification type and the information completed in the Information Exchange event profile.

Notification log information

The following is the format of information written to the Expedite Notification Manager notification log. Only parameters with nonblank values are written.

EVENT record

The syntax of the EVENT record is:

EVENT

```
EVENTDATE(date) EVENTTIME(time)
EVENTTYPE(S|M|A)
EVENTID(eventid)
ACCOUNT(account) USERID(userid)
SNDSYSID(sender's system ID) SNDACCOUNT(sender's account)
SNDUSERID(sender's userid)
ALIAS(alias type) ALIASNAME(alias name)
SENDERREF(reference ID) CLASS(class)
PRIORITY(A|H) MSGNAME(message name)
MSGLENGTH(message length) MODE(T/S/R/blank)
DATATYPE(A|E|N) MSGKEY(message key)
CYCLE(D|E|M|N|W|X|Y|Z) DAY(day)
TIME(time) TIMEZONE(offset)
TPNAME(transaction)
XTPNAME(hex transaction)
PARTNERLU(partner LU)
USERAPPL(user application)
APPLDATA(application data);
```

EVENTDATE

Date the event was received by Expedite Notification Manager. The format of this parameter is MM/DD/YYYY. This parameter contains 10 alphanumeric characters.

EVENTTIME

Time the event was received by Expedite Notification Manager. The time is displayed in local system time and military time. The format of this parameter is HH:MM:SS. This parameter contains 8 alphanumeric characters.

EVENTTYPE

Type of notification received.

- S** Scheduled event
- M** Message-arrival event
- A** Administrator/user-initiated event

EVENTID

Event ID specified by the user for the event. This parameter contains 1 to 8 alphanumeric characters.

ACCOUNT

Account for whom the event occurred. This parameter contains 1 to 8 alphanumeric characters.

USERID

User ID for whom the event occurred. This parameter contains 1 to 8 alphanumeric characters.

SNDSYSID

System ID of the message sender. This parameter contains 1 to 3 alphanumeric characters.

SNDACCOUNT

Account ID of the message sender. This parameter contains 1 to 8 alphanumeric characters.

SENDUSERID

User ID of the message sender. This parameter contains 1 to 8 alphanumeric characters.

ALIAS

Table type and table name of an alias table used by the message sender to refer to the receiver.

- G XXX** Global alias table, where XXX identifies a 1- to 3-character table name.
- O XXX** Organizational alias table, where XXX identifies a 1- to 3-character table name.
- P XXX** Private alias table, where XXX identifies a 1- to 3-character table name.

This parameter contains 1 to 4 alphanumeric characters.

ALIASNAME

Alias name defined in the alias table used by the message sender to refer to the receiver. This parameter contains 1 to 16 alphanumeric characters.

SENDERREF

Sender ID specified in the EDI data, if applicable. This parameter contains 1 to 35 characters.

CLASS

User class of the data specified by the message sender to identify the data. This parameter contains 1 to 8 alphanumeric characters.

PRIORITY

Message priority specified in the event definition.

A All messages

H High-priority

MSGNAME

Name for the data specified by the message sender. This parameter contains 1 to 8 alphanumeric characters.

MSGLength

Message size in characters, without headers and CDH. This parameter contains 1 to 10 numeric characters.

MODE

Sender-specified network data class field for this data.

blank A normal message with no CDH

T A test-mode message with no CDH

S A normal message with CDH available

R A test-mode message with CDH available

DATATYPE

Data type selected.

A Any data

E EDI data

N Non-EDI data

MSGKEY

Message key for specific receive. This parameter contains 1 to 20 alphanumeric characters.

CYCLE

Scheduled-event cycle definition.

D Daily

E Last day of the month

M Monthly

N Every day except the day specified in the DAY field.

W Weekly

X Every day except Saturday

Y Every day except Sunday

Z Every day except Saturday and Sunday

DAY

Days of the week or month in the definition that triggered the event. This field defaults to blanks for message-arrival events. For M and N schedules, this field contains the exact date (1-31). For schedules of D, X, Y, Z, E, this field contains 000. For W schedules, this field contains a cumulative value for all days.

Day of the week:	Assigned value:	Bit expression:
Sunday	128	1... ..
Monday	64	.1.. ..
Tuesday	32	..1. ..
Wednesday	16	...1 ..
Thursday	8 1..
Friday	41..
Saturday	21.
reserved	11

This is a cumulative number representing the sum of all scheduled days. For example, d=%DAY% substitution value displays as d=112. This represents the total of 64 + 32 + 16, or Monday, Tuesday, and Wednesday. This parameter contains 1 to 3 numeric characters.

TIME

Scheduled time in the definition that triggered the event. The time is based on a 24-hour clock. For example, 09:00=9 a.m.; 16:00=4 p.m.; 21:00=9 p.m. The scheduled time displays in the time zone of the scheduled-event definition owner. This parameter is in the format HH:MM and contains 5 numeric characters.

TIMEZONE

Event offset from Greenwich mean time in the format EHHMM or WHHMM. This parameter contains 5 alphanumeric characters.

TPNAME

TPNAME as defined in the Information Exchange Administration Services event profile. The object by the same name must be defined in the AS/400 QGPL. This parameter contains 1 to 64 alphanumeric characters.

XTPNAME

Hexadecimal representation of the TPNAME, if it was specified as a hex value in Information Exchange Administration Services. This parameter contains 1 to 128 alphanumeric characters.

PARTNERLU

Partner's LU name defined in the Information Exchange Administration Services event profile. This parameter contains 1 to 8 alphanumeric characters.

USERAPPL

User application field as defined in the Information Exchange Administration Services event profile. This represents the program that is to be submitted on the user's Expedite Notification Manager system. This parameter contains 1 to 75 alphanumeric characters; however, only the first 21 characters are used because of file naming limitations.

APPLDATA

Application data field specified in the Information Exchange Administration Services event profile. This parameter contains from 1 to 300 alphanumeric characters. This data is used as the call parameters of the submitted USERAPPL.

RETURN record

The syntax of the RETURN record is:

```
RETURN(return) REASON(reason)  
ERRTEXT(error text) ...;
```

RETURN

Completion code for the Expedite Notification Manager command. If the return code is zero, the command completed normally. If the return code is not zero, Expedite Notification Manager displays an error number and possibly a REASON and ERRTEXT value. This parameter contains 1 to 5 numeric characters.

REASON

More information about what caused the error. Expedite Notification Manager usually includes REASON when an error occurs while opening a file. This parameter contains 1 to 76 alphanumeric characters.

ERRTEXT

Detailed description of an error and may suggest steps to correct the problem. Expedite Notification Manager may include this parameter more than once in a record if the error text consists of multiple lines. This parameter contains 1 to 76 alphanumeric characters.

Notification log examples

The following are examples of Expedite Notification Manager notification logs:

- User-initiated

```
EVENT
EVENTDATE(02/14/1997) EVENTTIME(14:07:42)
EVENTTYPE(A) ACCOUNT(ACCT) USERID(USER1) TPNAME(ENMGR)
PARTNERLU(ACCTN03)
USERAPPL(1ERUNCLP) APPLDATA(ACCT USERG);
RETURN(0);
```

- Message-arrival

```
EVENT
EVENTDATE(02/14/1997) EVENTTIME(14:08:00)
EVENTTYPE(M) ACCOUNT(ACCT) USERID(USER1) SNDACCOUNT(ACCT)
SNDUSERID(USR2)
CLASS(FFMSG001) MSGLENGTH(000000079) MODE(S) DATATYPE(N)
MSGKEY(FFAD163464279E96F4F4) TPNAME(ENMGR) PARTNERLU(ACCTNM03)
USERAPPL(QGPL/IEMSG) APPLDATA(msgkey=FFAD163464279E96F4F4);
RETURN(0);
```

- Scheduled

```
EVENT
EVENTDATE(02/14/1997) EVENTTIME(14:14:43)
EVENTTYPE(S) EVENTID(SCHED) ACCOUNT(ACCT) USERID(USER1) CYCLE(D)
DAY(000) TIME(14:14) TIMEZONE(W0400) TPNAME(ENMGR) PARTNERLU(ACCTM03)
USERAPPL(USERPGM) APPLDATA(cycle=D);
RETURN(0);
```


Application Examples

Using application data or notification log data, you can develop applications that perform event-unique functions. The examples provided here run Expedite 400 Base using account and user IDs provided at run-time based on event data received from Information Exchange. All source code examples provided here are available in ENM400/SAMPLESRC.

Application example 1

The Information Exchange Administration Services profile for USERG is shown below. The account and user ID for whom the notification is sent is passed as application data. This data is used to run Expedite 400 Base for USERG's profile and library.

```

M1471US      Add or Change a User's Event Profile - Call Out

  Account ID / User ID.... ATAP      / USERG

  From - To time allowed for message arrivals..... 00 : 00 - 23 : 59
  Schedule delay..... 00 : 00 (HH : MM)
  Archives/system errors as msg arrivals? (Y/N)..... Y / Y
  Expedite Notification Manager format? (Y/N)..... Y
  Partner LU.....
ENM400LU
  Is TPNAME character or hexadecimal? (C/X)..... C
  TPNAME.... ENMGR
  User application.....
  IERUNCLP
  Application data..... %ACCOUNT% %USERID%

```

This profile's events will invoke Expedite Notification Manager program ENMGR on your AS/400. ENMGR will then submit program IERUNCLP PARM(ATAP USERG) to the job queue for execution.

Application example 2

The account and user ID for whom the notification is sent (USERG in this example) is parsed from the event notification log. This data is used to run Expedite 400 for USERG's profile and library.

```
M1471US          Add or Change a User's Event Profile - Call Out
Account ID / User ID.... ATAP      / USERG
From - To time allowed for message arrivals..... 00 : 00 : 59
Schedule delay..... 00 : 00 : MM)
Archives/system errors as msg arrivals? (Y/N)..... Y / Y
Expedite Notification Manager format? (Y/N)..... Y
Partner LU.....
ENM400LU
Is TPNAME character or hexadecimal? (C/X)..... C
TPNAME.... ENMCLP
User application.....
IEREXXCLP
Application data.....
```

This profile's events will invoke program ENMCLP on your AS/400. ENMCLP overrides the notification log file ENMNTFY so that the notification data is written to a member with the name ENMXXXXXX, where XXXXXX is the 6-digit job number. ENMCLP then calls Expedite Notification Manager program ENMGR. Expedite Notification Manager always submits user applications with the job name set to be ENMXXXXXX, where XXXXXX is the 6-digit job number under which ENMGR is running. User application program IEREXXCLP is submitted to the job queue by ENMGR. Program IEREXXCLP starts the REXX procedure IEREXX. IEREXX parses the account and user ID for whom the notification was sent from the notification log. IEREXX then calls program IERUNCLP (described in example 1) passing the account and user ID as parameters.

Expedite Notification Manager logs and traces

When you need detailed information about Expedite Notification Manager processing, you can examine the Expedite Notification Manager logs and traces. The notification log is always generated when notifications are received. To generate any run-time or trace information, you need to specify a trace command in your profile command file. This appendix describes the logs and traces, and provides examples.

Expedite Notification Manager logs

There are two types of Expedite Notification Manager logs: the notification log (ENMNTFY) and the run log (ENMRUN). The notification log documents the received notification, and the run log documents Expedite Notification Manager processing.

Notification log

Expedite Notification Manager places all notification data in the notification log. The Expedite Notification Manager notification log contains the data from notifications, such as event date, event time, event type, final return-code data, and possible error text. For an example of the Expedite Notification Manager Notification log, see Chapter 5, "Receiving notification data."

Run log

During execution, Expedite Notification Manager creates various messages that display the status of Expedite Notification Manager. These messages can be written to the file referenced by the ENMRUN by specifying LOG(Y) on the TRACE command, or they can be suppressed completely by specifying LOG(N).

The following is an example of the run-time information written to the file.

```
I Starting ENM on Fri Feb 14 10:53:36 1997
I Connected
I Receiving notification
I Decoding GDS
I GDS decoded
I Sending acknowledgment
I Disconnecting
I Calling IEREXXCLP
I Ending ENM
```

Expedite Notification Manager traces

There are two types of Expedite Notification Manager trace files: the link trace (ENMLINK) and the logic trace (ENMTRACE). These trace files are used for problem determination. All trace file options default to N, so the appropriate trace parameter must be set to Y on the TRACE command for the trace files to be created.

Link trace - TRACE LINK(Y)

The link trace lists the flow of data across the communication link. This trace is used primarily for problem determination by GXS Community Support.

```
Expedite Notification for AS/400 Version 1.1 - 02/14/97 started - Wed Jun 26 12:
12:40:53 | CMAACP(Accept Conversation) CONVID(73C8) Return(OK)
12:40:53 | CMECT(Extract Conversation Type) CONVID(73C8) Type(Mapped) turn(OK)
12:40:54 | CMEMN(Extract Mode Name) CONVID(73C8) Mode(LU62SYS1)
Return(OK)
12:40:54 | CMEPLN(Extract Partner LU) LU(IBINET.IBM0DIM1 ) Return(OK)
12:40:54 | CMESL(Extract Sync Level) Level(None) Return(OK)
12:40:54 | CMECS(Extract Conversation State) State(Receive) Return(OK)
12:40:54 | CMSRT(Set Receive Type) RECEIVE(Wait) Return(OK)
12:40:54 | CMRCV(Receive) CONVID(73C8) Complete(Yes) Send(Yes) Return(OK)
12:40:55 | CMSEND(Send) CONVID(73C8) Sent(000000) Return(OK)
12:40:55 | CMRCV(Disconnect) Return(Normal)
```

Logic trace - TRACE LOGIC(Y)

The logic trace lists the logic flow between Expedite Notification Manager components. This trace is primarily used for problem determination by GXS Community Support. The trace is stored in the file ENMTRACE.

Logic trace example:

```
=Message=>Expedite Notification for AS/400 started - Version 1.1 - 02/14/97<-
=Module=>ProTrace rc(0)<-
=Module=>outretn rc(0)<-
=Module=>checkrc rc(0)<-
=Module=>getrc rc(0)<-
=Module=>outprorc rc(0)<-
=Module=>ProProfile rc(0)<-
=Message=>Opening file as >text stream<-
=Message=>ENMERROR<-
=Message=>open mode->r<-
=Message=>Open successful<-
=Message=>Opening file as >text stream<-
=Message=>ENMRUN<-
=Message=>open mode->a<-
=Message=>Open successful<-
=Module=>ENMinit rc(0)<-
=Message=>Opening file as >text stream<-
=Message=>ENMERROR<-
=Message=>open mode->r<-
```

```
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMRUN<-  
=Message=>open mode->a<-  
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMERROR<-  
=Message=>open mode->r<-  
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMRUN<-  
=Message=>open mode->a<-  
=Message=>Open successful<-
```

Logic Trace example

```
=Module=>ENMrcvn rc(0)<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMERROR<-  
=Message=>open mode->r<-  
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMRUN<-  
=Message=>open mode->a<-  
=Message=>Open successful<-  
=Module=>valgds rc(0)<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMNTFY<-  
=Message=>open mode->a<-  
=Message=>Open successful<-  
=Module=>DecodeNLog rc(0)<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMERROR<-  
=Message=>open mode->r<-  
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMRUN<-  
=Message=>open mode->a<-  
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMERROR<-  
=Message=>open mode->r<-  
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMRUN<-  
=Message=>open mode->a<-  
=Message=>Open successful<-  
=Module=>ENMsndr rc(0)<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMERROR<-  
=Message=>open mode->r<-  
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-
```

```
=Message=>ENMRUN<-  
=Message=>open mode->a<-  
=Message=>Open successful<-  
=Module=>Disconnect rc(0)<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMERROR<-  
=Message=>open mode->r<-  
=Message=>Open successful<-  
=Message=>Opening file as >text stream<-  
=Message=>ENMRUN<-  
=Message=>open mode->a<-  
=Message=>Open successful<-
```

NOTE: A job end code of 0 does not necessarily mean that Expedite Notification Manager ran without errors. Check the ENM400 files, OUTQ and MSGQ, for additional information.

Messages and codes

This appendix lists and describes messages that Expedite Notification Manager generates. Informational messages are preceded by the letter I, error messages are preceded by the letter E, and warning messages are preceded by the letter W.

19022 | **Receiving notification**

Explanation: The program is attempting to receive a notification.

User Response: Informational only. No response is needed.

19023 | **Decoding GDS**

Explanation: A notification has been received and is being decoded.

User Response: Informational only. No response is needed.

19024 | **GDS decoded**

Explanation: The notification has been successfully decoded.

User Response: Informational only. No response is needed.

19025 | **Sending acknowledgment**

Explanation: Information Exchange is being given notification status.

User Response: Informational only. No response is needed.

19026 | **Calling %s**

Explanation: The program is attempting to run the application passed in the received notification.

User Response: Informational only. No response is needed.

19027 | **Disconnecting**

Explanation: The program is being disconnected from Information Exchange.

User Response: Informational only. No response is needed.

19029 | Connected

Explanation: The application is establishing a communication session with Information Exchange.

User Response: Informational only. No response is needed.

19051 | Starting ENM on %s

Explanation: The Expedite Notification Manager program is starting.

User Response: Informational only. No response is needed.

19052 E Error allocating memory

Explanation: The program could not allocate enough memory to perform a necessary function.

User Response: End all unnecessary programs in the system.

19056 E Error in ENM profile

Explanation: An error occurred while processing the ENMIPRO file.

User Response: Check prior messages and ENMOPRO for possible causes.

19057 E Error during execution of user application

Explanation: The operating system reported an error while attempting to execute the user application.

User Response: Check prior messages for possible causes.

19101 E Invalid profile command

Explanation: The profile command file contained an invalid profile command.

User Response: Check your profile command file ENMIPRO for possible causes.

19102 E Invalid profile parameter

Explanation: A profile command contained an invalid parameter.

User Response: Check your profile command file ENMIPRO for possible causes.

19103 E Missing semicolon or expected parameter not found

Explanation: Expedite Notification Manager did not get the expected parameter or semicolon.

User Response: Check your profile command file ENMIPRO for possible causes.

19104 E Expected profile command not found

Explanation: Expedite Notification Manager did not find the expected profile command.

User Response: Check your profile command file ENMIPRO for possible causes.

19105 E Duplicate profile parameter found

Explanation: A duplicate parameter was encountered.

User Response: Check your profile command file ENMIPRO for possible causes.

19106 E NULL character found in profile file

Explanation: A NULL character was found in your command profile file.

User Response: Check your profile command file ENMIPRO for the NULL character and remove any occurrences.

19107 E Profile command or parameter too long

Explanation: A command or parameter was encountered that is too long.

User Response: Check your profile command file ENMIPRO for possible causes.

19131 E Invalid USERAPPL on EXECUTE command

Explanation: You specified an invalid USERAPPL parameter in the EXECUTE command. USERAPPL must be 0 or 1.

User Response: Correct the USERAPPL parameter on the EXECUTE command in the profile command file ENMIPRO. Retry the program.

19132 E Invalid APPLDATA on EXECUTE command

Explanation: You specified an invalid APPLDATA parameter in the EXECUTE command. APPLDATA must be 0 or 1.

User Response: Correct the APPLDATA parameter on the EXECUTE command in the profile command file ENMIPRO. Retry the program.

19134 E USERAPPL required with APPLDATA on EXECUTE command

Explanation: You specified APPLDATA in the EXECUTE command without providing the USERAPPL parameter.

User Response: Correct the USERAPPL parameter in the EXECUTE command in the profile command file ENMIPRO. Retry the program.

19141 E Invalid LINK specified on TRACE command

Explanation: You specified an invalid LINK parameter on the TRACE command. LINK must be Y or N.

User Response: Correct the LINK parameter on the TRACE command in the profile command file ENMIPRO. Retry the program.

19142 E Invalid LOGIC specified on TRACE command

Explanation: You specified an invalid LOGIC parameter in the TRACE command. LOGIC must be Y or N.

User Response: Correct the LOGIC parameter on the TRACE command in the profile command file ENMIPRO. Retry the program.

19143 E Invalid LOG specified on TRACE command

Explanation: You specified an invalid LOG parameter in the TRACE command. LOG must be Y or N.

User Response: Correct the LOG parameter on the TRACE command in the profile command file ENMIPRO. Retry the program.

19156 E Error calling user's application

Explanation: The user application could not be run.

User Response: Check prior messages for possible causes.

19157 E Application, %s, invoked with return code of %s

Explanation: The user application executed and ended with a non-zero return code.

User Response: Check the user application documentation for the meaning of the returned code.

19177 E Invalid conversation type

Explanation: An incorrect conversation type was found between the Expedite Notification Manager and Information Exchange.

User Response: Contact the Customer Assistance Department. You will need to provide an ENMLINK trace file.

19178 W Unable to determine conversation type

Explanation: Expedite Notification Manager was unable to determine conversation type.

User Response: This information will not be included in the ENMLINK trace file.

19179 W Unable to determine mode name

Explanation: Expedite Notification Manager was unable to determine the mode name.

User Response: This information will not be included in the ENMLINK trace file.

19180 W Unable to determine partner LU

Explanation: Expedite Notification Manager was unable to determine the partner LU.

User Response: This information will not be included in the ENMLINK trace file.

19181 W Unable to determine sync level

Explanation: Expedite Notification Manager was unable to determine the sync level.

User Response: This information will not be included in the ENMLINK trace file.

19182 E Invalid sync level

Explanation: The sync level between Expedite Notification Manager and Information Exchange is incorrect.

User Response: Contact the Customer Assistance Department. You will need to provide an ENMLINK trace file.

19183 W Unable to determine conversation state

Explanation: Expedite Notification Manager was unable to determine the conversation state.

User Response: This information will not be included in the ENMLINK trace file.

19185 E Unable to set receive type

Explanation: Expedite Notification Manager was unable to set the receive type.

User Response: This information will not be included in the ENMLINK trace file.

19201 E Error receiving notification

Explanation: An error occurred while attempting to receive a notification.

User Response: Check prior messages for possible causes.

19251 E Error decoding gds in notification

Explanation: An invalid notification GDS was received.

User Response: Check file ENMTRACE for possible causes.

19252 E Error opening notification log

Explanation: The notification log could not be opened.

User Response: The file may be in use by another program. End all other applications and try again.

19253 E Error in notification log data

Explanation: An error occurred while updating the notification log.

System Action: The file may be in use by another program or the file may be out of space. End all other applications, process any unprocessed notifications, reallocate the file and try again.

19254 E Error writing notification log

Explanation: An error occurred while updating the notification log.

User Response: The file may be in use by another program or the file may be out of space. End all other applications, process any unprocessed notifications, reallocate the file and try again.

19256 E Error in GDSID from IE

Explanation: The data received was not recognized as a valid notification.

User Response: No action may be necessary. This may happen if your Information Exchange event profile specified a value of N for Expedite Notification Manager format. Please change that value to a Y and try again.

19257 E Error in GDS data from IE

Explanation: The notification received has an internal error.

User Response: No action may be necessary. If the problem continues, notify the Customer Assistance Department.

19258 E Error in GDS length from IE

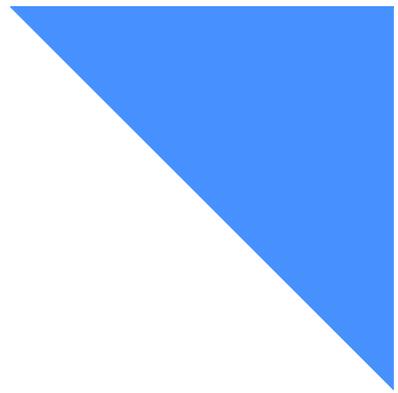
Explanation: The notification received has an internal error.

User Response: No action may be necessary. If the problem continues, notify the Customer Assistance Department.

99999 | Ending ENM

Explanation: Expedite Notification Manager is ending.

User Response: Informational only. Check prior messages to determine if Expedite Notification Manager completed successfully.



Glossary

This glossary defines words as they are used in this book. This glossary includes terms and definitions from:

- The American National Dictionary for Information Systems, ANSI x3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American Standards Institute, 11 West 42 Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The Information Technology Vocabulary, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition.

A

account. A name that identifies an account to a program, device, or system.

alphanumeric. Generally, any keyboard character, but for practical purposes should be restricted to alphabetic, numeric, space, and common punctuation characters.

C

CDH. Common Data Header.

character. Generally, any letter, number, punctuation mark, or other symbol that can be entered at a computer keyboard.

command. A request to run a particular program or function.

Common Data Header (CDH). A set of control information about a file, which is sent to Information Exchange by the sending interface. When the file is received by the trading partner, the receiving interface can use the information in the CDH.

Customer Assistance Department. The group or organization within GXS that responds to customer questions and assists them in using the product or service. See also Help Desk Operator.

D

default. An alternative value an application program uses automatically when none has been specified.

E

EBCDIC. Extended binary-coded decimal interchange code.

error message file (ENMERROR). The file that provides Expedite Notification Manager with error descriptions in the RETURN and PROFILER output records.

ENMTEXT. Extended error text file.

event. Any action that generates a notification in Information Exchange.

ENM. Expedite Notification Manager.

Expedite Notification Manager. An Advantise-developed application that resides on the user's system and either answers the phone for dial-out, or is in the application started for LU62 leased-line call-out.

G

GDS. Generalized Data Stream.

Generalized Data Stream (GDS). A format of data as it comes from Information Exchange.

H

Help Desk Operator. A person who receives questions or problem reports from network users. See also Customer Assistance Department.

I

Information Exchange Base (IEBASE) program. The program that starts Expedite Base/400.

Information Exchange. A communication service that permits users to send and receive information electronically.

Information Exchange Administration Services. An online, panel-driven product that the Information Exchange Service Administrator uses to perform administrative tasks for Information Exchange.

Information Exchange mailbox. See mailbox. Information Exchange service administrator. The person who coordinates the use of Information Exchange within a company.

M

mailbox. A temporary storage area for electronic mail from which data is retrieved by the intended recipient. message-arrival event. The arrival of a message that meets criteria defined in the message-arrival definition.

P

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which you specify a value or for which the system provides a value when the menu is interpreted. (3) Data passed between programs or procedures.

profile command file (ENMIPRO). A file in which you place profile commands pertaining to profile commands.

profile response file (ENMOPRO). A file containing Expedite Notification Manager's response to profile commands.

response file. An output file into which Expedite Notification Manager echoes commands from ENMIPRO along with its associated return codes.

S

scheduled event. An event that occurs on a specified schedule.

service administrator. A primary contact person in an organization who controls use of the service by the users within the organization and who assists the service support groups.

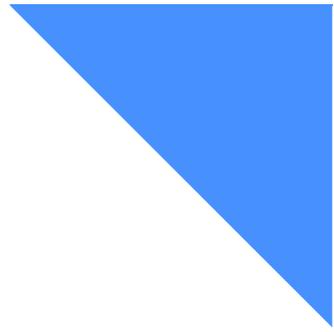
syntax. The rules for constructing a command.

T

trace file. A file that provides you with trace information of transactions. In Expedite Notification Manager there are two types of trace files: link trace (ENMLINK) and logic trace (ENMTRACE).

U

user-initiated event. An event initiated by the user to occur immediately or at a specified date and time. The event occurs only once.



Index

A

attributes
ENMIPRO 7
ENMLINK 8
ENMNTFY 6
ENMOPRO 7
ENMRUN 9
ENMTEXT 8
ENMTRACE 8

C

codes and messages 33
command syntax conventions v

E

ENMERROR 8
ENMIPRO
attributes 7
creating profile 11
example 7
profile commands 11
ENMLINK
attributes 8
ENMNTFY
attributes 6
example 6
ENMOPRO
attributes 7
ENMRUN
example 9
ENMTEXT 8
errdesc 14
error message file (ENMERROR) 8

events

message arrival 1
scheduled 1
user-initiated 1
EXECUTE command 12
Expedite Notification Manager
files 5
introducing 1
logs and traces 27
notification log 5
optional files 7
receiving notifications 17
system diagram 2
extended error text file (ENMTEXT) 8

F

files, optional 7

I

Installing Expedite Notification Manager 3

L

link trace 29
link trace file (ENMLINK) 8
logic trace 29
logic trace file (ENMTRACE) 8
logs
notification log 28
run log 28

M

message-arrival event 1
messages and codes 33

N

notification log
 ddname (ENMNTFY) 5
 examples 23
 how used 5
notification log information
 EVENT record 18
 RETURN record 22
notifications, receiving 17

O

optional files
 8
 ENMIPRO 7
 ENMLINK 8
 ENMOPRO 7
 ENMRUN 9
 ENMTEXT 8
 ENMTRACE 8

P

profile command file (ENMIPRO) 7
profile commands
 execute 12
 trace 13
profile commands, Expedite Notification
 Manager 11
profile response file (ENMOPRO) 7

R

receiving notifications 17
run log 28
run-time information (ENMRUN) 9

S

scheduled event 1
syntax conventions, command v
system diagram 2

T

TRACE 13
TRACE command example 13
traces
 command example 13
 link 13
 link trace 29
 log 13
 logic 13
 logic trace 29

U

user-initiated event 1