



iSeries

System Manager Use

Version 5

SC41-5321-02





@server

iSeries

System Manager Use

Version 5

SC41-5321-02

Note

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

Third Edition (August 2002)

This edition replaces SC41-5321-01. This edition applies only to reduced instruction set computer (RISC) systems.

© Copyright International Business Machines Corporation 1997, 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
--------------------------	------------

Notices	ix
Programming Interface Information.	x
Trademarks.	x

About System Manager Use (SC41-5321)	xiii
---	-------------

Who should read this book	xiii
Conventions and terminology used in this book	xiii
Prerequisite and related information	xiii
iSeries Navigator	xiv
How to send your comments	xiv

Summary of Changes	xv
-------------------------------------	-----------

Chapter 1. Setting Up for System Manager **1**

Preparing to set up your servers.	1
Supporting systems at different release levels	4
Planning for storage space.	6
Mapping the steps for setting up	6
Using Advanced Peer-to-Peer Networking (APPN)	7
Distributing PTFs using change management	8
Naming the systems in your network	8
Using alert support.	10
Distribution using SNADS	13
Verifying contact information	14
Setting up a server as a service provider.	14
Installing Managed System Services on a service provider	14
Configuring Managed System Services	14
Starting Managed System Services.	15
Installing System Manager on a service provider	16
Configuring System Manager on a service provider	16
Security considerations	17
Setting up service support for a service provider	18
Setting the service provider attributes	18
Setting up the list of service requester servers	19
Setting up product support	22
Working with supported products (WRKSPTPRD command).	22
Starting System Manager	25
Starting Manager Services	26
Setting up a server as a service requester and a managed system.	26
Installing Managed System Services on a service requester	26
Starting Managed System Services.	26
Setting up a list of service providers	27
Resolving problems.	29
Verifying the network setup	30
Example networks	31

iSeries example network	31
Setting up System Manager	32
System/370 and iSeries Example Network	34
Setting up System Manager with Host System/370	34

Chapter 2. Product packaging concepts **37**

Prepackaging considerations.	37
Marketing and delivery	37
The application environment	38
Maintenance	39
Development Process	40
National language support	41
Dynamic naming	41
Dynamic naming restrictions	42
Multiple releases and repackaging considerations.	42
Packaging libraries, folders, and directories.	43
Product development, testing, and usage modes	43
Establishing a development environment for directories	45
Storing document library objects in auxiliary storage pools	45
Working with database files	46
Packaging a follow-on release	46
Repackaging considerations	46

Chapter 3. Packaging your products . . . **47**

OS/400 release considerations	48
Product packaging restrictions	48
Step 1: Creating a product definition	49
Product example: Creating a product definition	50
Step 2: Creating a product load.	52
Product example: Creating a product load	54
Packaging products that have more than one library	55
Packaging products that have folders and documents.	56
Example of packaging a product by using folders	57
Packaging products that have directories	61
Understanding national language loads for your product.	67
Step 3: Packaging products that have exit programs	72
Preoperation exit program	73
Postoperation exit program	73
Step 4: Creating message files and messages	78
Step 5: Marking the objects for packaging	78
Product example: Marking objects for packaging	78
Step 6: Packaging your product option	80
Product example: Packaging your product option	82
Step 7: Saving and restoring your product	85
Saving Your License Program	85
Restoring your license program.	85
Step 8: Checking your product structure.	86
Product packaging summary	87
Commonly asked product packaging questions	89

Chapter 4. Packaging products for licensing 93

Overview of license management	93
Planning	94
Ordering process changes	94
License term strategy	94
Type of compliance regulations to enforce	95
Usage types	95
Release considerations	96
Packaging user-based priced products	96
Defining features	96
Adding license information	98
License information options	101
Creating a license key	102
Application program interface (API) calls	102
Requesting a use of the license	103
Releasing a use of the license	103
Compliance considerations	104
Authority considerations	105
Requesting a use of the license examples	105
Concurrent use example	105
Registered use example	107

Chapter 5. Product installation, distribution, and management 111

Saving licensed programs	111
Installing a packaged product	112
Avoiding library name conflicts at installation time	112
Testing an installed product	112
Distributing your product	113
Sending a Product	113
Installing a remote product	113
Product backup and recovery considerations	113
Problem analysis for an installed product	113

Chapter 6. Using the program temporary fix (PTF) functions 115

Overview of change management on an OS/400 service provider	115
Work with supported products (WRKSPTPRD) command	115
Ordering program temporary fixes (PTFs)	116
Ordering the PTFs that you need	116
Automatically ordering the PTFs that you need from IBM service support	119
Special PTF notifications	120
Receiving program temporary fixes (PTFs)	121
Checking for prerequisite and corequisite PTFs	122
Receiving PTFs from IBM service support electronically	122
Receiving PTFs from OS/400 service providers electronically	122
Receiving PTFs from distribution media	123
Working with PTF save files	124
PTF cover letters	125
Working with cover letters	125
Work with PTF commands	127
Work with Program Temporary Fixes (WRKPTF) Command	127
Working with assigned PTF IDs	127

Change the PTF range	128
Display Program Temporary Fix (DSPPTF) command	128
Testing PTFs before distribution	131
Releasing previously held PTFs for distribution	132
Deleting save files, cover letters, and on-order status	132
Cleaning up unnecessary PTF objects	133
Distributing PTFs	133
Sending PTFs to another site	133
Working with PTFs remotely	135
Creating PTF packages	136
Cumulative PTF packages	137

Chapter 7. Servicing your products using System Manager 141

Service information for the product owner	141
Getting your system ready to create PTFs	141
Creating a program temporary fix	142
Assigning an identification name to your program temporary fixes	142
Establishing libraries for your program temporary fixes	143
Replacing and adding objects to your product using program temporary fixes	143
Establishing dependencies on other PTFs	143
Preparing PTF cover letters	144
Searching the problem log	145
Using program temporary fixes to change documents inside folders	146
Using program temporary fixes to change stream files inside directories	146
Developing PTF exit programs to eliminate manual operations	147
Displaying details about a program temporary fix	148
Applying a program temporary fix	148
Testing PTFs	148
Rebuilding PTFs	149
Backup and recovery of PTF information	149
Distributing PTFs	149
Product user information for PTF developers	150
Analyzing problems	150
Installing PTFs	151
Saving and restoring PTFs	152
Cleaning up PTF information	152

Chapter 8. Working with the Problem Log. 153

Using the problem log commands	157
Work with Problem (WRKPRB) command	157
Display Problems (DSPPRB) command	159
Change Problem (CHGPRB) command	161
Delete Problem (DLTPRB) command	162
Working with a problem	164
Analyzing a problem	164
Reporting a problem	165
Running recovery procedures for a problem	165
Verifying a problem is corrected	165
Answering a problem	165

Sending a response for a problem	166
Closing a problem	166
Collecting authorized program analysis report (APAR) data	166
Displaying authorized program analysis report (APAR) data	167
Save authorized program analysis report (APAR) data	167
Delete authorized program analysis report (APAR) data	167
Understanding problem status changes	167
OPENED status	167
READY status	168
PREPARED status	169
SENT status	170
ANSWERED status	171
VERIFIED status	172
CLOSED status	173
Status changes and operations	173
Tracking problems	175
Problem filtering	176

Chapter 9. Detecting, analyzing, and reporting remote problems 179

Detecting problems	180
System-detected problems	180
Application-detected problems	180
User-detected problems	181
Receiving problem notifications	181
Automatic problem analysis	182
Reporting problems detected by the system	182
Service requests as problem notifications	183
Alerts as problem notifications	183
Running remote problem analysis	184
Starting remote problem analysis	184
Completing problem analysis	187
Possible causes lists	187
Answering problems	188
Software problems	189
Hardware problems	192
Reporting problems from the service provider	192
Problem analysis commands	194
Work with Problem (WRKPRB) command	194
Analyze Problem (ANZPRB) command	194
Work with Alerts (WRKALR) command	194
Verify Communications (VFYCMN) command	195
Verify Tape (VFYTAP) command	195
Problem management APIs	195

Chapter 10. Controlling operations centrally 197

Creating a change request description	198
Adding activities to change request descriptions	199
Scheduling activities	201
Conditioning	202
Node list or node names	202
Standard names versus global names	203
Text Description	203
Defining your own activity types	204
Exit program data	205

How exit programs are called	205
Displaying CRQ description activity details	205
Displaying nodes	206
Displaying conditions	206
End code definitions	207
Displaying type specific data	208
Submitting change requests	208
What happens if the request does not run	208
Activity considerations	209
Tracking submitted change requests	210
Working with submitted change requests	210
Combining more than one activity into one change request description	218
Printing the change request description	219
Submitting the revised change request	219
Accessing fast path distribution commands	219
Managed systems node list (NODL)	222
Managed system node names (CPNAME)	222
Sending a license	223
Supported OS/400 object types for object activities	224
Distributing installable objects	225
Using global names with CRQ activities	225
Using component names with CRQ activities	226
Commands associated with change requests	227
Cleaning the change request database	228
Using commands on remote systems	229
Displaying output	230
Diagnosing problems with change request processing	230
System Manager is not started	230
Managed System Services is not started	231
QSNADS subsystem is not started at the central site system	231
SVDS distribution queue is not created	231
Attempt to send remote command failed	231

Appendix A. Programming and Command Reference 233

Appendix B. Example setup and configuration of an OS/400 network using SNADS 235

Using Systems Network Architecture distribution services (SNADS)	235
Creating directory entries	235
Setting up SNADS for object distribution	235
Verifying SNA distribution services (SNADS) configuration	237
Creating directory entries on the service requester	238
Creating directory entries on the service provider	239
Matching parameter values for setting up the iSeries servers	239
Setting up communications on the service provider	240
Changing the network attributes	241
Line, controller, and device descriptions on the service provider	241
SNADS definitions on the service provider	242
Setting up the list of service requesters	244
Setting up communications on the service requester	247

Changing the network attributes for a service requester	248
Line, controller, and device descriptions on service requesters	248
SNADS definitions on the service requester	249
Setting up the service requester for OS/400 service provider support	250
Setting up the list of service providers	250
Creating a user profile for remote problem analysis	252
Granting object authority on the service requester	253
Changing system values on the service requester	253
Setting up another service requester	253
Example work sheet	254

Appendix C. Examples of problem situations 257

General assumptions about attributes	257
Alert reported a system-detected problem	257
Service request reported a system-detected problem	259
User-detected performance problem	260
PTF order and distribution	262
Detailed examples of problem situations	263
Hardware service request report	263
User-detected performance problem	266
Application-detected field value problem	268
Alert reported printer problem	269
Send PTF Order (SNDPTFORD) from a service requester	270
When things do not work correctly	272
Problem records not created when alerts are received	272
Problem records not created when service requests are received	272
Cannot find messages on the service provider	273
PTFs not sent	273
PTFs sent but never arrive	274
PTF applied by service provider does not solve the problem	274
Takes a long time to add product support by using Work with Supported Products command	275
Cannot do remote problem analysis	275
Service requester is not entitled	275

Appendix D. Example product exit programs 277

Appendix E. System Manager APIs and exit programs 291

Change Request APIs	292
Add Change Management Activity (QNSADDCM) API	292
Authorities and Locks	293
Required Parameter Group	293
Optional Parameter Group	297

Usage Notes	297
Error Messages	297
Send Change Request Message (QNSSNDCM) API	303
Authorities and Locks	303
Required Parameter Group	303
Error Messages	304
Update Status (QNSUPDST) API	306
Authorities and Locks	306
Required Parameter Group	306
Error Messages	307
Add Activity Exit Program	308
Required Parameter Group	309
Notes	309
Error Messages	310
Change Activity Exit Program	310
Required Parameter Group	310
Error Messages	310
Display Activity Exit Program	311
Required Parameter Group	311
Error Messages	313
Run Activity Exit Program	314
Required Parameter Group	314
Error Messages	316
Action Value List	316
Parameter Descriptions	323
Other APIs—Used with Products	340

Appendix F. Commands shipped with public authority *Exclude 343

Appendix G. Authority required for objects used by commands 345

Appendix H. Authority to System Manager commands and APIs 349

Appendix I. User profile database file 355
User profile database logical file (QASVNUP) 355

Appendix J. QSVSM user profile default values 369

Bibliography 371

Bibliography	371
Systems Management	371
Software Management	371
Communications and Network Planning	371
System Operation	371
National Language Support	371
iSeries Access information	371
NetView Distribution Information	372
Backup and Recovery	372
Security Information	372

Index 373

Figures

1. Map of Commands for Setting Up System Manager	7
2. An iSeries Network with System Manager Installed	9
3. Two iSeries Networks	32
4. Service Providers Network Attributes for Alert Support.	32
5. Service Provider Attributes	33
6. SYSA1 and SYSA2 Network Attributes for Alerts	33
7. A System/370 or System/390 with an iSeries Network	34
8. Service Providers Network Attributes for Alert Support.	35
9. Creating a Product Example, Step 1	51
10. Product Example, Step 2	54
11. Packaging a Product with Folders	61
12. Packaging a Product with Directories	62
13. Product Objects for a Product Installed on Two Systems.	68
14. Product Example, Step 3	79
15. Product Example, Step 4	83
16. Defining Features Using the CRTPRDDFN and CRTPRDLOD Commands.	97
17. Adding License Information	99
18. Application Program Interface (API) Calls	103
19. IBM Query Output.	131
20. Operations Available for a Problem in OPENED Status.	168
21. Operations Available for a Problem in READY Status	169
22. Operations Available for a Problem in PREPARED Status	170
23. Operations Available for a Problem in SENT Status	171
24. Operations Available for a Problem in ANSWERED Status	172
25. Operations Available for a Problem in VERIFIED Status	173
26. Communications Flow for Problem Management.	180
27. Central Site System and Managed Systems	194
28. Send SMG Object (SNDSMGOBJ)	222
29. An Example OS/400 Network	238
30. Matching Command Parameter Values	240
31. Communications Flow for Alert Reported Problem	258
32. Service Request Reported System-Detected Problem	259
33. A User-Detected Performance Problem	261
34. A Send PTF Order and Distribution	262
35. Operations for System-Detected Hardware Problem	264
36. Operations for a User-Detected Performance Problem	267
37. Operations for a Printer Problem	269
38. Operations to Send a PTF Order and Distribution	271
39. Example of Action Value List	322

Notices

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

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your area or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator
3605 Highway 52 N
Rochester, MN 55901-7829
U.S.A.

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

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

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

This information 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.

COPYRIGHT LICENSE:

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

Programming Interface Information

This publication is intended to help you to complete systems management tasks. This publication documents General-Use Programming Interface and Associated Guidance Information provided by System Manager for iSeries licensed program.

General-Use programming interfaces allow the customer to write programs that obtain the services of System Manager for iSeries licensed program.

Trademarks

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

Advanced Function Printing
Advanced Peer-to-Peer Networking
Advanced 36
AFP

AIX
AnyNet
Application System/400
APPN
AS/400
AS/400e
C/400
e (logo)
iSeries
iSeries 400
IBM
NetView
Operating System/2
Operating System/400
Operational Assistant
OS/2
OS/400
PS/2
Redbooks
RISC System/6000
RS/6000
SAA
SystemView
System/36
System/370
System/38
System/390
Systems Application Architecture
S/370
S/390
WebSphere
400

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other locations, or both.

UNIX is a registered trademark of The Open Group in the United States and other locations.

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

About System Manager Use (SC41-5321)

This book provides a description of how to:

- Set up and manage a network of systems including application software management, data distribution, centralized operations, and centralized problem management.
- Package and manage your own products.

This book assumes that you have installed your systems and configured your network using Advanced Peer-to-Peer Networking (APPN). It is also assumed that you have installed the System Manager licensed program on each appropriate system in your network.

Who should read this book

This book is intended for the following groups of users:

- Application developers who want to package and manage their own products in the same manner IBM products are managed.
- Central site network operators responsible for planning and maintaining a network of IBM servers.

An understanding of communications configuration, APPN, and Systems Network Architecture distribution services (SNADS) is assumed.

Conventions and terminology used in this book

This book refers to packaging a product as the process of converting your application into a product that contains hierarchical structures. Packaging allows your product to exist in library, folder, or directory structures. Product home directories and product directories are new for the directory types to which you are packaging your product.

Prerequisite and related information

Use the iSeries Information Center as your starting point for looking up iSeries and AS/400e technical information. You can access the Information Center two ways:

- From the following Web site:
<http://www.ibm.com/eserver/iseries/infocenter>
- From CD-ROMs that ship with your Operating System/400 order:
iSeries Information Center, SK3T-4091-02. This package also includes the PDF versions of iSeries manuals, *iSeries Information Center: Supplemental Manuals*, SK3T-4092-01, which replaces the Softcopy Library CD-ROM.

The iSeries Information Center contains advisors and important topics such as Java, TCP/IP, Web serving, secured networks, logical partitions, clustering, CL commands, and system application programming interfaces (APIs). It also includes links to related IBM Redbooks and Internet links to other IBM Web sites such as the Technical Studio and the IBM home page.

With every new hardware order, you receive the *iSeries Setup and Operations CD-ROM*, SK3T-4098-01. This CD-ROM contains IBM @server iSeries Access for

Windows and the EZ-Setup wizard. iSeries Access offers a powerful set of client and server capabilities for connecting PCs to iSeries servers. The EZ-Setup wizard automates many of the iSeries setup tasks.

For related information, see the “Bibliography” on page 371.

iSeries Navigator

IBM iSeries Navigator is a powerful graphical interface for managing your iSeries and AS/400e servers. iSeries Navigator functionality includes system navigation, configuration, planning capabilities, and online help to guide you through your tasks. iSeries Navigator makes operation and administration of the server easier and more productive and is the only user interface to the new, advanced features of the OS/400 operating system. It also includes Management Central for managing multiple servers from a central system.

For more information on iSeries Navigator, see the iSeries Information Center.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other iSeries documentation, fill out the readers’ comment form at the back of this book.

- If you prefer to send comments by mail, use the readers’ comment form with the address that is printed on the back. If you are mailing a readers’ comment form from a location other than the United States, you can give the form to the local IBM branch office or IBM representative for postage-paid mailing.
- If you prefer to send comments by FAX, use either of the following numbers:
 - United States, Canada, and Puerto Rico: 1-800-937-3430
 - Other countries or regions: 1-507-253-5192
- If you prefer to send comments electronically, use one of these e-mail addresses:
 - Comments on books:
RCHCLERK@us.ibm.com
 - Comments on the iSeries Information Center:
RCHINFOC@us.ibm.com

Be sure to include the following:

- The name of the book or iSeries Information Center topic.
- The publication number of a book.
- The page number or topic of a book to which your comment applies.

Summary of Changes

| Here is what has changed since V5R1:

- | • CL command descriptions and syntax diagrams moved to Information Center
| (<http://www.ibm.com/eserver/iseriess/infocenter>)
- | • The chapters and appendixes that documented the System Manager GUI
| function were removed.
- | • In V5R1 the password parameter changed to allow an increase in the password
| length from 10 to 128 characters for the following System Manager commands:
 - | – Add Command Change Request Activity (ADDCMDCRQA)
 - | – Change Command Change Request Activity (CHGCMDCRQA)
 - | – Run System Manager Command (RUNSMGCMD)

| If the iSeries server is running with a password level of 2 or 3, the password
| coming from the source iSeries server will not be known on the target system.

- | • The OS/400 licensed program and iSeries Navigator provide support that allows
| you to manage PTF groups (or fix groups). At this time, the System Manager
| licensed program does not support PTF groups.
- | • The QALWOBJRST system value enables security-sensitive objects to be restored
| during a PTF install when you specify *ALWPTF.

Chapter 1. Setting Up for System Manager

This chapter provides information and procedures to help you set up System Manager on the servers in your network. With the System Manager licensed program, you can organize systems management tasks to suit your business needs.

The needs of your business change, sometimes sooner than you expect. To respond to business changes effectively, your servers must change too. Managing your servers, at first glance, might seem like just another time-consuming job. But the systems management investment pays off soon in the form of an efficient server. It is efficient because changes are planned and managed. This means you can control operations from one site, a central site, within a network. Using operations management products, being in control of servers no longer means that you have to be at those servers; you can control many operations on many servers from one place. Centralized control of operations also enables better tracking of functions performed on the server.

Products that manage OS/400 operations centrally include:

- System Manager for iSeries (5722-SM1)
- Managed System Services for iSeries (5722-MG1)

A **central site system** is the system in the network that controls network operations. From the central site system, skilled operators control the other servers in the network. These other servers are called managed systems. A **managed system** is a system that is managed by operators at the central site.

The **service provider** is a central site system that handles all the problems for the network. The **service requester** is any system in the network that has a problem and requires service. An OS/400 business computing system with the System Manager licensed program installed has the ability to be a service provider for a group of service requesters in a network. More than one OS/400 service provider may be in a network.

For System Manager service providers to manage resources, they must interact with a set of service requesters. A service requester is a system with a program or equipment problem that requires and asks for problem-handling support from another system in the network that is acting as a service provider. To request service from an OS/400 service provider, a system must be defined on the service provider as a service requester. An iSeries server is a service requester whenever the operator or the system sends an alert, a service request, or a PTF order.

Preparing to set up your servers

Before you set up System Manager, you need to verify that you have enough disk storage space to store the PTF save files you need to provide service and support, that communications are properly set up, and that the contact information on each system in the network is current. The following sections describe the aspects of software release level compatibility, storage space, and communications that you need to consider.

Before you set up System Manager, you need to look at the characteristics of your server and your network. The following items that you need to consider are described in the chapters of this book.

- It is assumed that you have installed the System Manager licensed program before you begin using the procedures for setting it up. See “Installing System Manager on a service provider” on page 16 for additional information about installing the System Manager program.
- You need Managed System Services installed at the central site system to use the distribution and installation functions. See “Installing Managed System Services on a service requester” on page 26 for additional information about installing the Managed System Services program.
- A service provider must be running the same or higher release level of the Operating System/400[®] program as the service requesters. This enables the service provider to provide support for service requesters running at the same release level or a previous release level.
- Verify that the appropriate release level is supported.
- Evaluate the storage requirements
- Consider what is necessary to use Advanced Peer-to-Peer Networking[®] (APPN[®]). System Manager is dependent on the use of the Advanced Peer-to-Peer Networking (APPN) function for its service provider and service requester support.

Using the AnyNet[®] support provided in OS/400, System Manager can perform software distribution and installation in TCP/IP networks as well.

- Configure the central site system

Managed system attributes and configuration values that affect the entire operation of the managed system must be specified before you begin using Managed System Services. The central site system and all systems in the network must install the Managed System Services licensed program to allow communication within the network.

- Determine how your network uses alert support

Another aspect of problem management you should look at before you set up System Manager is whether or not you are currently using OS/400 alert support. **Alerts** are records sent to a focal point to identify a problem or an impending problem.

- Set up your network to use Systems Network Architecture distribution services (SNADS)

System Manager uses SNADS to distribute software and to control software installation. System Manager is dependent on some SNADS requirements. A SystemView[®] Distribution Services (*SVDS) distribution queue is needed on the central site system for each managed system that is supported. This queue is created automatically when the managed system is added as a service requester. If you do not configure a managed system as a service requester, then a queue will need to be added manually using the Add Distribution Queue (ADDDSTQ) command.

You use the SystemView Distribution Services to send and receive distributions to managed systems in a network from a central site system. Each managed system should be defined as if it was directly connected to a central site system either logically or physically.

- Configuring supporting products

The Work with Supported Products (WRKSPTPRD) command allows you to specify the list of products, product options, and language features for which you are willing to process service requests and PTF orders. Using the OS/400

program a service provider can provide support only for certain releases. You can use the Release (RLS) parameter to specify the release with which you want to work.

As new products are purchased and installed by your service requesters, you must use this command on your service providers to indicate if they will support (supply service for) this new product. This is also true as new release levels of products or product options are installed by your service requesters. It is a good idea to run this command as new licensed programs are installed to ensure product support for your network of systems is current and correct. Also, if you want to be able to receive PTFs at your service providers, the products they are associated with must be either installed or supported.

- Verify the contact information

The final preparation before you set up System Manager is to verify both your local contact information and the contact information on the remote servers in your network. **Contact information** is included in a problem record when you send a service request or PTF order. Contact information is used by the service provider (or IBM® service support) to identify the source of the request. The contact information includes the name, mailing address, and telephone number of the person to contact at your location when a problem occurs. You can also use the Work with Contact Information (WRKCNTINF) command on the OS/400 system. See “Verifying contact information” on page 14 for more information.

- Verify that System Manager is ready to use by using Send Service Request (SNDSRVRQS) command with the *TEST parameter to verify that the systems are set up correctly.
- Verify that Managed System Services is configured on the managed systems in your network.

An iSeries server, acting as a service requester only, does not need to have System Manager installed to request service from an OS/400 service provider. However, both systems must be properly configured for communications, using Advanced Peer-to-Peer Networking (APPN) and SNADS, and be set up so that the service provider can effectively use the functions provided by the System Manager licensed program.

Additional preparation information for setting up your systems using SNADS is in Appendix B, “Example setup and configuration of an OS/400 network using SNADS” on page 235.

System Manager is a user-based priced product. With System Manager a user is a remote server that is managed. These include remote iSeries servers, remote servers such as RISC System/6000® servers, and remote workstations such as PS/2's. For more information about user-based priced products, see Chapter 4, “Packaging products for licensing” on page 93.

Table 1 shows the products required for each task you want to do.

Table 1. Required OS/400 Products by Task

Task	OS/400® Products Required		
	Central Site System	Managed System	System Manager
	Managed System Services	Managed System Services	
Package applications into products	X		

Table 1. Required OS/400 Products by Task (continued)

Task	OS/400® Products Required		
	Central Site System	Managed System	System Manager
	Managed System Services	Managed System Services	
Report problems	X		
Track problems	X		
Analyze problems on managed systems	X		
Distribute PTFs	X	X	X
Distribute products	X	X	X
Prepare products and package installable objects for distribution		X ¹	X ¹
Distribute files	X	X	X
Install products	X	X	X
Load and apply PTFs	X	X	X
Run programs on managed systems	X	X	X
Perform IPL to managed systems	X	X	X
Run commands on managed systems	X	X	X
Install installable objects	X	X	X
Uninstall installable objects	X	X	X
Distribute license keys	X	X	X
Note:			
¹ Managed System Services can be on either the central site system or the managed system.			

Supporting systems at different release levels

If you are upgrading your systems to the current release of the Operating System/400 program and installing System Manager, you may need to do some backup and recovery procedures on the service provider, both before and after the installation of the new release. See the *Software Installation* book for more complete information about upgrading the systems in your network and installing the System Manager licensed program.

System Manager allows a service provider to perform problem management and change management tasks for service requesters that are running either the current or a previous release of the OS/400 program. The exception to this is a change in the modification level of the operating system.

Table 2 on page 5 shows the release level requirements supported by service providers for the OS/400 licensed program. The service and support for the service

requester is provided by an System Manager service provider running a certain release of System Manager.

Table 2. Service Provider Release Support Table

Service Provider Release Level	V4 R4 M0	V4 R5M0	V5 R1 M0	V5 R2 M0
V4 R4 M0	X	X	X	X
V4 R5 M0	X	X	X	X
V5 R1 M0	X	X	X	X
V5 R2 M0	X	X	X	X
Note: X = Releases that are supported - = Releases that are not supported				

Note: If your server is the service provider and you intend to support servers running both the current and previous releases of the OS/400 program, your server must be at the highest release level of all the servers in the network.

To control the network efficiently, it is suggested that you use Managed System Services with System Manager. When you configure the managed systems that have Managed System Services installed, you use configuration values that affect the entire operation of the managed system. These values specify:

- Whether to accept or reject activities received from the central site system.
- The security program that is run to determine which activities to accept, which user profile to use when processing activities, and which authorization list to use to secure distribution repository objects.
- The default user profile to use when processing activities.
- The time-out values for jobs waiting to process activities.
- Whether to send intermediate responses to the central site system.
- The prefix tokens that specify how automatic global-to-local name mapping should be done.
- The remote command attributes such as:
 - Remote command security program
 - Maximum return data
 - Delete spooled file
 - Remote command key

The user profile on the central site system that has authorization to the user profile database may be able to view user profile information for user profiles that he or she does not have authority to view on the actual systems where the user profile resides. To ensure proper security, you have the capability to encode passwords when working with user profiles.

You can use a journal to monitor changes to the file, if you are concerned about changes to the user profile database.

A distribution queue of type *SVDS is required for each central site system from which the managed system will receive requests.

Planning for storage space

System Manager assumes you will store program temporary fix (PTF) save files on the service provider. These files must be kept for a period of time to provide effective service. Adequate disk space is required to store the PTF save files.

The best way to minimize the amount of disk space used by PTF save files is to be sure all the systems in your network have applied the latest cumulative PTF package before setting up System Manager. Doing this assures you that all your systems have the most current level of software installed and lessens the number of PTFs you need to order for corrective maintenance. If you follow this recommendation, you should allow approximately 20MB of disk storage for the anticipated IBM PTF save files for a single release. You must also anticipate the amount of disk storage required for fixes (PTFs) for application software. If you do not apply the latest cumulative PTF package to all the servers in your network, you may require more than this amount of disk space. See “Overview of change management on an OS/400 service provider” on page 115 for more information about cumulative PTF packages and change management strategy.

If you have a large number of service requesters for which you intend to provide support, the service provider problem log and alert log can become large. Periodic maintenance, such as closing and deleting problems and deleting alerts, is necessary to conserve storage.

To clear the disk space, you need to either use the cleanup functions available with the Operational Assistant function (type GO MANAGESYS on any command line), or you can periodically reorganize the physical files that are associated with the problem and alert logs. For more information about problem log maintenance, see “Delete Problem (DLTPRB) command” on page 162. For more information about alert log maintenance, see “Maintaining the alert log” on page 12.

Mapping the steps for setting up

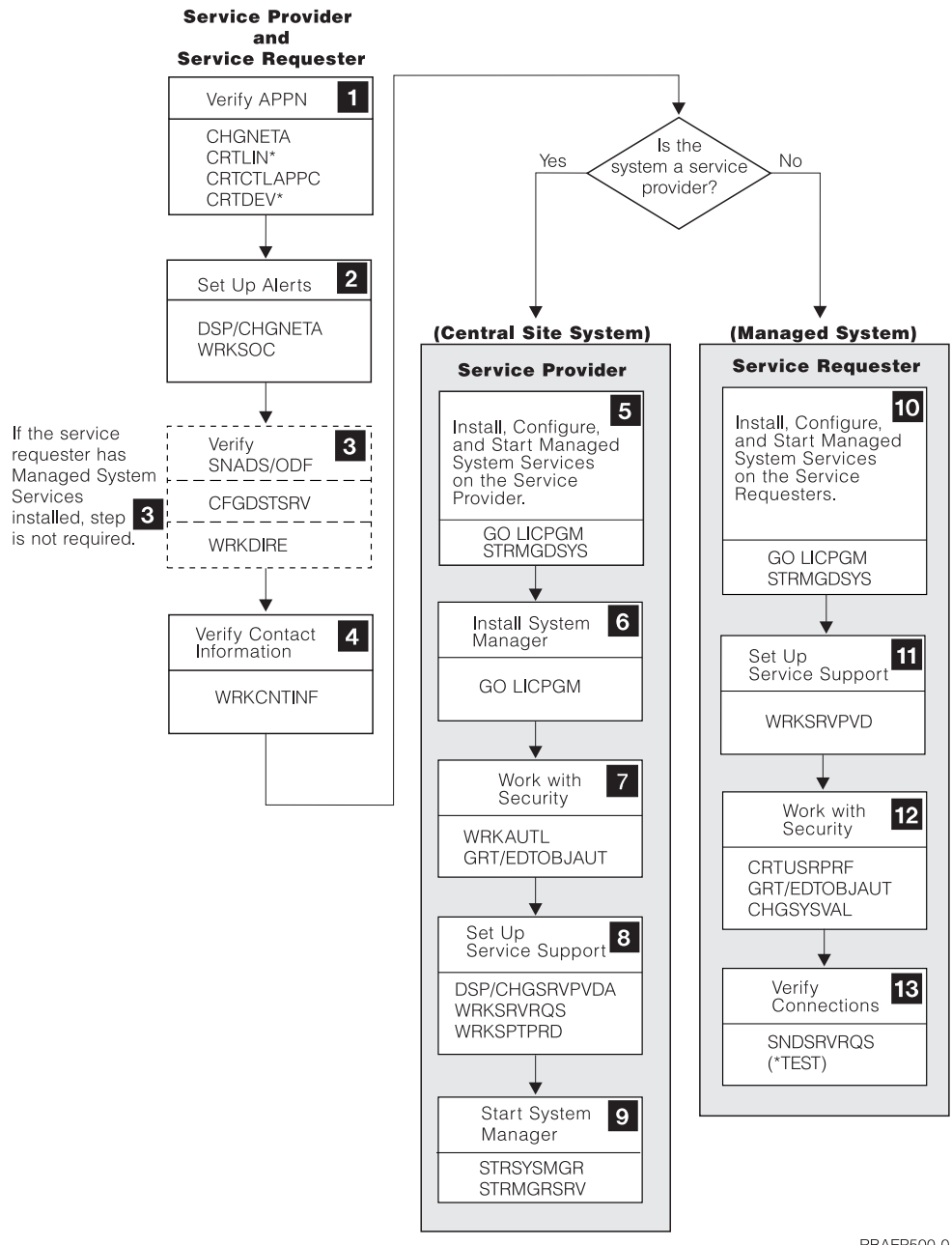
The map of commands shown in Figure 1 on page 7 is an overview showing the commands you may use when preparing for and setting up System Manager. Follow steps 1 through 4 to set up communications and alert support. These topics are covered in depth in the Information Center:

- APPN support, which is available in the Networking topic.

- Alerts support, which is available in the Networking topic.

Follow steps 5 through 9 to set up System Manager. The step numbers are noted in reverse image in the text with the corresponding procedure.

An example of the commands and displays used to set up an OS/400 network and the System Manager licensed program is provided in Appendix B, “Example setup and configuration of an OS/400 network using SNADS” on page 235. Throughout this chapter, references are made to this example so that you can look at the procedures in a network environment as you set up your systems.



RBAEP500-0

Figure 1. Map of Commands for Setting Up System Manager

Using Advanced Peer-to-Peer Networking (APPN)

System Manager is dependent on the use of the Advanced Peer-to-Peer Networking (APPN) function for its service provider and service requester support. APPN provides data communications support that routes data in a network between two or more systems that do not need to be adjacent. For a system to provide effective service, connectivity between the service provider and service requesters in the network must be possible using APPN. If the service provider is going to run remote problem analysis and distribute PTFs to a service requester, the service provider must be able to call the service requester.

Before you begin setting up a system as a service provider, you should check some of the characteristics of the system and the network. Because the use of APPN is assumed, you need to have appropriate line and controller descriptions according to the communications protocol you are using. See “Line, controller, and device descriptions on the service provider” on page 241 and “Line, controller, and device descriptions on service requesters” on page 248 for examples of setting up communications in an iSeries network. For more information on setting up these descriptions, see the Networking topic in the Information Center for the following information units:

- APPN
- Communications configuration
- ISDN support
- LAN, Frame-relay, and ATM support
- X.25 network support

You may also need to update the network attributes and possibly the APPN remote configuration list that is associated with your system. These checks correspond with step 1, in Figure 1 on page 7.

The APPN mode #INTER is used for communications between a service requester and a service provider, specifically for sending service requests and performing remote problem analysis. The APPN mode #INTER is shipped with the iSeries server and must be configured on all the systems in the network. If your iSeries server communicates through a network control program (NCP), you must add #INTER to the network control program log mode table.

Note: It is recommended that you use line descriptions other than the ones shipped for communications with IBM service support (QESLINE) and IBM Marketing support (QTILINE) for communications between the service provider and the service requesters. You should create separate communications objects (line, controller, and device descriptions) even if the same physical line will be used.

For switched line configurations, you must vary on the communications line and controller before sending a service request from a service requester to a service provider.

Distributing PTFs using change management

Once your system is configured, you should decide how to handle PTFs. You can use the fast path commands to distribute PTFs or create a change request description to do several operations and schedule them at specific times. For more details about the commands, see “Accessing fast path distribution commands” on page 219. For more details about using a change request description, see Chapter 10, “Controlling operations centrally” on page 197.

With the System Manager licensed program, you can send program temporary fixes (PTFs) to managed systems and retrieve PTFs from the managed systems. You also have the option to apply, remove, and delete remote PTFs.

Naming the systems in your network

To simplify management of your network, you should follow a naming pattern for the names of systems, network IDs, control points, and locations. Looking at the topology of your network can help with this. To avoid confusion when trying to identify which system or system description you are working with, use the same

name for the system name, APPN control point name, and local location name. For example, Figure 2 shows a large network composed of two network groups. Each system has a local location name, a system name, and an APPN control point name. The name for the service provider in the network shown is HQSYS. The names for the service requesters shown are CHICAGO1, ACCTSYS, and MFGSYS. These names should be unique within a network ID.

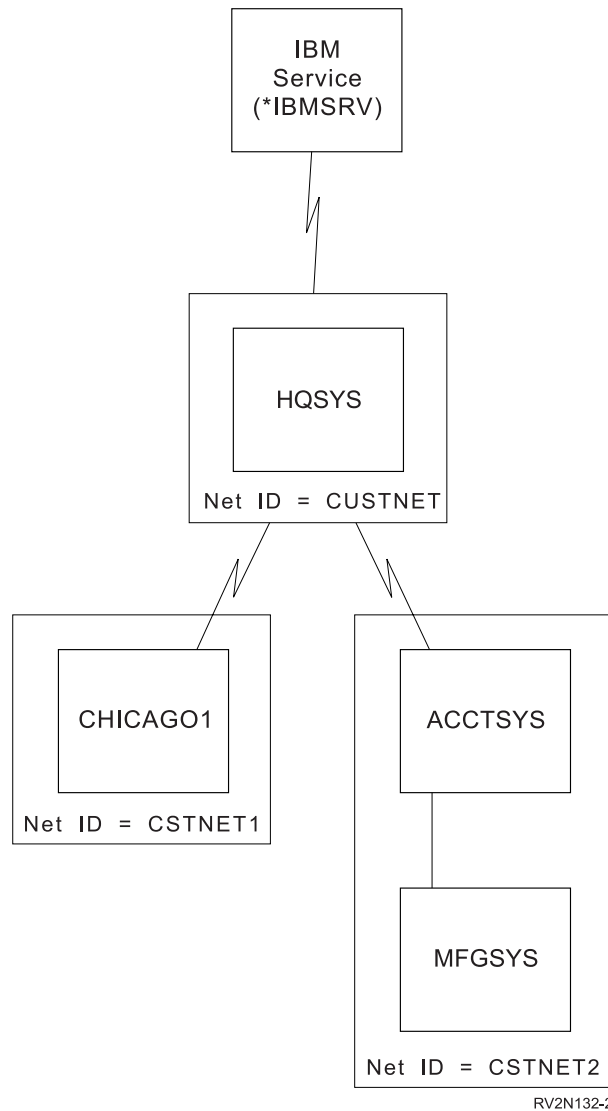


Figure 2. An iSeries Network with System Manager Installed

In Systems Network Architecture (SNA), a network is an interconnected group of systems that are logically associated by a common identifier called a network ID or net ID. Network IDs should be consistent within an SNA network. A network ID is an identifier composed of 1 to 8 alphanumeric characters. The combination of the network ID and the control point name provides a unique name for each system in a large network consisting of several smaller networks. Communications between SNA networks require that each network ID be unique. In the example network shown in Figure 2 on page 9, three distinct network IDs are shown: CUSTNET, CSTNET1, and CSTNET2.

Previously, the network owner was responsible for assigning the network ID and ensuring its uniqueness. IBM established the SNA network registry to assist network owners in maintaining the uniqueness of their network IDs.

The primary reason for registering a network ID is to establish its uniqueness. IBM encourages all network owners to register their network IDs; however, registration is a voluntary decision. A registered network ID consists of 8 alphanumeric characters. Use of 8-character network IDs that are not registered is not recommended. For more information about registering your network ID, contact your IBM branch office representative.

Tip

The network ID in the node name uses the local system network ID as the default. You only need to specify the network ID if the node in the node list entry is in a different APPN network.

Verifying your system names

The names you choose for the systems in your network must be reflected in the network attributes for both service providers and service requesters. To verify the network attributes for system names, do the following:

1. Type the Display Network Attributes (DSPNETA) command on a command line and press F4 (Prompt).
2. Verify the current system name (SYSNAME).
3. Verify the local network ID (LCLNETID). This name should be consistent within an SNA network.
4. Verify the local control point name (LCLCPNAME) and the default local location name (LCLLOCNAME).

An example showing the command to change the network attributes for system names is in “Changing the network attributes” on page 241.

Note: The current system name (SYSNAME), local control point name (LCLCPNAME), and the local location name (LCLLOCNAME) do not have to be the same; however, using the same name for each keyword value simplifies communications configuration and setup for System Manager.

Using alert support

Another aspect of problem management you should look at before you set up System Manager is whether or not you are currently using OS/400 alert support. **Alerts** are records sent to a focal point to identify a problem or an impending problem. Alerts provide notification of system-detected problems occurring on remote systems in the network. This is especially valuable if some or all of the remote systems in the network do not have system operators maintaining them.

An alert is sent from the system where the problem has been detected to a system assigned as an alert focal point. An **alert focal point** is a system in a network that receives and processes (logs, displays, and optionally forwards) alerts from other systems in the network. The group of systems that send alerts to the same focal point is called that focal point’s sphere of control. The **sphere of control** of a system is a collection of control points for which that system is acting as a focal point. This collection includes both control points explicitly defined by the customer if the controlling system is a primary focal point, control points assumed

by the system if the controlling system is a default focal point, and control points assumed by the system if the controlling system is a backup or requested focal point.

Note: If all the service requesters in your network are end nodes, you do not need to designate the service provider as an alert focal point to have alerts received there. You can designate the service provider as the network node by setting the node type parameter (NODETYPE) to *NETNODE using the Change Network Attributes (CHGNETA) command. Then have the service requesters specify the service provider as the network server by setting the network node servers parameter (NETSERVER) to the network ID and control point name of the service provider using the Change Network Attributes (CHGNETA) command.

If you have already set up alert support in your network, you should have your alert focal point set up as your central site system (service provider). Doing this allows the operator at the central site system (service provider) to create a problem record from the alert, either manually or automatically. You can then use the problem record to start remote problem analysis and track and answer remote problems from a central site in the network.

Note: Use the Work with Problem (WRKPRB) command to ensure that the alert caused a problem record to be created on the service requester. If an alert on the service requester lacks a problem record, the alert cannot be used to create a problem record on the service provider.

Setting up basic alert support

The following procedures assist you with the setup for basic alert support. These procedures correspond to step 2, in Figure 1 on page 7.

To make a service provider an alert focal point, type on any command line:

```
CHGNETA ALRSTS(*ON) ALRPRIFP(*YES) ALRLOGSTS(*ALL)
```

Note: If you do not want the service provider to create alerts, set the alert status (ALRSTS) attribute to *OFF.

To turn alerts on at your service requesters, have an operator at each service requester type on any command line:

```
CHGNETA ALRSTS(XXX)
```

The XXX in the command may be either *UNATTEND, if the system is not normally attended by an operator, or *ON, if the system is normally attended by an operator who can run local problem analysis before reporting the problem to the service provider.

Note: If you are going to run remote problem analysis for system-detected problems from the service provider on the service requester, a problem record must exist for the problem on both systems. If the alert status network attribute is set to *UNATTEND at the service requester, an alert is created each time a problem record is created on the system. When you set up your service provider attributes, be sure to set the log remote problem (LOGRMTPRB) attribute to *ALERT or *ALL to log these problems at the service provider automatically. The problem records are very important for running remote problem analysis from the service provider on a service requester that is running without a full-time operator.

When the alert setup procedure is completed, you need to either add the service requesters to the service provider's sphere of control (unless all the service requesters are end nodes), or change the alert controller description (ALRCTLDD) network attribute at each service requester to route the alerts to the service provider. The alert controller description is used when you have systems in the network that are not iSeries servers and you want these servers to send or receive alerts, or when the servers in your network are directly connected by way of switched lines. For example, you may want the service provider to log alerts and then forward them to a System/370™ or System/390®.

To add service requesters to the service provider's sphere of control:

1. Type the Work with Sphere of Control (WRKSOC) command on the command line at the service provider and press the Enter key.
2. Use option 1=Add to add systems to the sphere of control.
3. Type the remote network ID and control point name of each service requester that you want to add.

Use the WRKSOC command at the service provider to display, change, or remove sphere of control entries.

OR

To change the alert controller description, do the following at each service requester:

1. Type the Work with Controller Descriptions (WRKCTLD) command on the command line and press the Enter key.
2. Locate the controller used for sessions between the service requester and the service provider. Note the controller name.
3. Type CHGNETA ALRCTLDD(controller name) and press the Enter key.

Note: For more information on how to set up and use the alert controller description, see the *Alerts Support* book.

Use the Work with Alerts (WRKALR) command to view and work with alerts received from the service requesters for which you provide support.

Maintaining the alert log

The average alert size is 300 to 350 bytes. If you are logging alerts at the service provider, the amount of disk space taken up can become large. Periodically, you need to delete alerts from the alert log. When you do this, it is also a good time to free the disk space used by the physical file associated with alert management. Use the automatic cleanup functions provided by the Operational Assistant function.

To set up the automatic cleanup functions, do the following:

1. Type GO SETUP on any command line and press the Enter key.
2. Select option 2 (Cleanup tasks). The Cleanup Tasks display is shown.
3. Select option 1 (Change cleanup) options to set up the automatic cleanup for your system.
4. Type the appropriate values for your system in the fields shown.

System cleanup can be scheduled to occur automatically every day, or you can manually start and end the cleanup functions. To be sure the disk space for alerts and problem records is cleared, set the *System journals and system logs* field to the number of days for which you want to keep alert and problem data.

For more complete information about setting up and using OS/400 alert support and alert log maintenance, see the *Alerts Support* book, which provides an introduction to alerts and describes all the commands and procedures necessary to set up more sophisticated alert support.

Alert filtering

Alert filtering is provided as a function of the alert manager that is a part of the Operating System/400 program. It is started when an alert is created or received.

Alert filtering consists of categorizing alerts into logical, manageable groups. Each alert group is assigned specific tasks or actions that are performed automatically, such as:

- Logging the alert in the alert log
- Assigning a user to the alert
- Sending the alert to another system
- Sending a notification record to a data queue

You can develop automatic alert handling applications using application program interfaces (APIs), such as the Retrieve Alert (QALRTVA) API. For example, your application can monitor a data queue to receive notification of an alert and then use the Retrieve Alert (QALRTVA) API to retrieve that alert.

The commands for working with alert filters are:

- Add Alert Action Entry (ADDALRACNE)
- Add Alert Selection Entry (ADDALRSLTE)
- Change Alert Action Entry (CHGALRACNE)
- Change Alert Selection Entry (CHGALRSLTE)
- Change Filter (CHGFTR)
- Create Filter (CRTFTR)
- Delete Filter (DLTFTR)
- Remove Filter Action Entry (RMVFTRACNE)
- Remove Filter Selection Entry (RMVFTRSLTE)
- Work with Filters (WRKFTR)
- Work with Filter Action Entry (WRKFTRACNE)
- Work with Filter Selection Entry (WRKFTRSLTE)

The Work with Alerts (WRKALR) command also has additional parameters for working with alert filtering.

Distribution using SNADS

Step 3 shows that distribution of requests and objects is performed using SNA distribution services (SNADS). The way in which SNADS is configured depends on whether or not the service requesters have installed the Managed System Services program. If a service requester has installed the Managed System Services program, the SNADS configuration is performed automatically. If the program is not installed, then some additional manual configuration is required. See Appendix B, "Example setup and configuration of an OS/400 network using SNADS" on page 235 for information about distribution using SNADS.

Verifying contact information

The final preparation before you set up System Manager is to verify both your local contact information and the contact information on the remote systems in your network. **Contact information** is included in a problem record when you send a service request or PTF order. It is used by the service provider (or IBM service support) to identify the source of the request. The mailing address of the enterprise is used to mail PTFs on media, or to dispatch a service representative to the correct location with replacement hardware parts. This is step 4, in Figure 1 on page 7. Use the following procedure on each system:

1. Type the Work with Contact Information (WRKCNTINF) command on the command line and press the Enter key.
2. Select option 2 (Work with local service information) from the Work with Support Contact Information menu.
3. Select option 2 (Change service contact information) from the Work with Local Service Information display.
4. Verify the name in the *Contact* field is the name of the person to contact at your location when a problem is sent from this system to a service provider.
5. Verify that the telephone numbers and mailing address are complete and correct. The telephone numbers should include necessary area codes, exchange numbers, and extensions. The mailing address is the address used for mailing PTF media and for a service representative to deliver a hardware replacement part.
6. Make any necessary corrections to the information and press the Enter key.

Notes:

1. This information may be changed for each service request at the time the service request is prepared for sending to a service provider.
2. All contact information is in the user's user profile entry. The contact information can be verified or changed on the Verify Contact Information display.

Setting up a server as a service provider

A server can be set up as a service provider for an entire network or group of networks. A service provider can report problems directly to IBM service support (*IBMSRV) or to a higher-level OS/400 service provider. All systems designated as service providers must have the System Manager licensed program installed.

Installing Managed System Services on a service provider

In step 5 you install, configure, and start Managed System Services on the service provider. See "Installing System Manager on a service provider" on page 16 for the steps to install System Manager.

1. Type `go licpgm` on any command line and press the Enter key. The Work with Licensed Programs menu is shown.
2. Select option 11 (Install licensed programs) and press the Enter key. The Install Licensed Programs display is shown.
3. Type a 1 (Install) in the *Opt* column next to the licensed program entry, 5722MG1 Managed System Services.

Configuring Managed System Services

During the installation of the Managed System Services licensed program, the program reads the list of service providers and creates a SystemView distribution

services (*SVDS) queue for each service provider (except *IBMSRV). This means that after the installation is complete, activities can be received from service providers immediately.

To display the Managed System Services menu, type `go svms` on any command line and press the Enter key. From this menu, you can configure Managed System Services to access the functions the program provides.

1. Select option 1 (Configure Managed System Services).
2. Select option 3 (Work with service providers).

From the Work with Service Providers display you can add, change, copy, remove, and display service providers. You can also press F13 (Configure distribution services) to set up your distribution queues or routing table.

Note: If Managed System Services is installed an *SVDS distribution queue is created automatically when a service provider is added using the Work with Service Provider (WRKSRVPVD) command.

You will need to create an *SVDS distribution queue. Use the F6 function key (Add distribution queue) to create this queue. You will need the following information to create the queue:

- Queue type
- Remote location name
- Local location name
- Queue
- Mode

If your central site system is not a service provider, use the Add Distribution Queue (ADDDSTQ) command to add a distribution queue for it. It is not necessary to configure an SNA distribution services (SNADS) routing table entry if the distribution queue name (DSTQ) is the same as the system network ID and control point name. For more complete SNADS configuration information, see the *SNA Distribution Services* book.

For information and detailed steps about how to set up service requesters, see “Setting up the list of service requester servers” on page 19.

Starting Managed System Services

The Start Managed System Services (STRMGDSYS) command starts jobs to perform managed system functions such as receiving objects, running programs, running commands, and applying program temporary fixes (PTFs). You need *JOBCTL authority to use this command. To start Managed System Services, type `strmgdsys` on any command line and press the Enter key.

The Starting Managed System Services allows the managed system to receive and process activities sent by the central site system.

Note: The QSNADS subsystem must be started for Managed System Services to be operational.

When you end Managed System Services, you can select the controlled option where you specify a delay time or the immediate option.

Installing System Manager on a service provider

System Manager must be ended before installing or saving it. This is done by issuing the End System Manager (ENDSYSMGR) command.

After the communications and object distribution has been set up, the network operator is ready to install the System Manager licensed program using the following steps. This procedure corresponds with step 6 in Figure 1 on page 7.

1. Type `go licpgm` on any command line and press the Enter key. The Work with Licensed Programs menu is shown.
2. Select option 11 (Install licensed programs) and press the Enter key. The Install Licensed Programs display is shown.
3. Type a 1 (Install) in the *Opt* column next to the licensed program entry, 5722SM1 System Manager.

Configuring System Manager on a service provider

To display the System Manager menu, type `go svsm` on any command line and press the Enter key. From this menu, you can configure System Manager to access the functions the program provides. Then, select option 1 (Configure System Manager) to do the following:

- Display service provider attributes
- Change service provider attributes
- Work with service requesters
- Work with node lists
- Configure distribution services

Using option 3 (Work with service requesters), you can add, change, display, or remove information about the service requesters within the network.

If you want to provide service and support for a remote system in your network, you must add that system to the list of service requesters shown on the Work with Service Requesters display. You need to enter the control point name, the network ID, and the description of the system you are adding. When you type a 1 in the Option column, a display is shown where you can add service contact information for the system.

Using a node list you can send a request to a list of systems. For more information about using a node list to send requests, see “Node list or node names” on page 202.

You can use option 5 (Configure distribution services) to identify the following:

- Distribution queues
- Routing table
- Secondary system name table

The *SVDS queue is automatically created when:

- A service provider is added or changed using the Work with Service Provider (WRKSRVPVD) command and Managed System Services is installed.
- A service requester is added or changed using the Work with Service Requester (WRKSRVRQS) command and *SVDS is specified.

Security considerations

When you first install the System Manager licensed program, some of the commands are set up with PUBLIC(*EXCLUDE) authority and others have PUBLIC(*USE) authority. Only users with all object (*ALLOBJ) authorization, such as the security officer (QSECOFR), can use the PUBLIC(*EXCLUDE) commands. An authorization list (AUTL) is created when System Manager is installed. The authorization list contains all the commands in the System Manager program that have the *EXCLUDE public authority and commands that are new to the system (meaning the commands that were not on the system when System Manager was installed). You can grant authority to the people in the list using the following procedure. This corresponds to step 7, in Figure 1 on page 7. For more information about how to add users to the authorization list as well as a sample program you can use, see Appendix G, “Authority required for objects used by commands” on page 345.

Note: The following procedures for granting authority to use the service provider commands are not the only procedures you can use. See the *iSeries Security Reference* book for other methods of granting authority to these commands.

1. Type the Work with Authorization Lists (WRKAUTL) command on any command line and press F4 (Prompt). Specify *ALL in the *Authorization list* field. The Work with Authorization List display is shown.
2. You can select option 1 (Create) an authorization list or select option 2 (Edit) to change the authorization list created during the installation of System Manager. Press the Enter key.
3. Press F5 (Refresh) to refresh the display. The new authorization list is shown.
4. Type a 2 (Edit) in the *Opt* column next to the name of the list you just created. Press the Enter key.
5. Press F6 (Add new users) to add users to the list.
6. Type the user IDs for the users you want to allow to use the service provider commands and specify *USE in the *Authority* field. Press the Enter key.
7. Press F3 (Exit) to leave the Work with Authorization List display. Now you can grant the authority for each command.

Appendix G, “Authority required for objects used by commands” on page 345 and Appendix H, “Authority to System Manager commands and APIs” on page 349 contain information about authorities and authorization lists.

The following procedure must be repeated for each of the System Manager CL commands that have PUBLIC(*EXCLUDE) authority.

1. Type the Grant Object Authority (GRTOBJAUT) command on a command line and press F4 (Prompt). The Grant Object Authority List display is shown.
2. Type a service provider command in the *Object* field. For example, type CHGSRVPVDA (library=QSYS).
3. Type *CMD in the *Object type* field.
4. Type the name of the authorization list you previously created in the *Authorization list* field. To include the commands that existed before System Manager was installed, add the QSM1AUTL to the authorization list. Press the Enter key.

A message appears letting you know that object authority has been changed.

For information about APIs, see Appendix E, “System Manager APIs and exit programs” on page 291 and for information about authorities, see Appendix H,

“Authority to System Manager commands and APIs” on page 349 and Appendix G, “Authority required for objects used by commands” on page 345.

You may need to set up an APPN remote location list if a location password is necessary to sign on to any of the remote systems for which you intend to provide service. You may also need to create an APPN remote location list if you are using switched lines to communicate with your remote systems. If your remote systems fall under either or both of these categories, you can use the Work with Configuration Lists (WRKCFGL) command to create (or add to) an APPN remote location list of type *APPNRMT. For more information about APPN communications security or the APPN remote location list, see the Networking topic in the Information Center.

Setting up service support for a service provider

The following procedures explain how to use the commands needed to set up your system to provide service for other iSeries servers in a network. These procedures correspond to step 8, in Figure 1 on page 7.

Setting the service provider attributes

1. Type the Change Service Provider Attributes (CHGSRVPVDA) command on any command line and press F4 (Prompt).
2. Type the values appropriate for your system in the space provided and press the Enter key.

See the display and explanation following this procedure for information about the individual service provider attributes.

The following display shows the default values for the parameters of the CHGSRVPVDA command.

```

Change Service Provider Attr (CHGSRVPVDA)

Type choices, press Enter.

Log remote problems . . . . . *SRVRQS      *SAME, *NONE, *SRVRQS...
Send PTFs automatically . . . . *NO      *SAME, *NO, *YES
Message queue . . . . . QSYSOPR      Name, *SAME
Library . . . . . QSYS      Name, *LIBL, *CURLIB
Maximum distribution . . . . . 000500    1-5000, *SAME, *NOMAX
Hold PTFs received . . . . . *YES      *SAME, *NO, *YES
Cover letter language . . . . . 2924    F4 for list
Forward requests automatically . *YES      *SAME, *NO, *YES

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

```

The service provider attributes let you decide how you receive problem notifications and PTF orders and how you distribute PTFs. The following list describes the service provider attributes.

Log remote problems (LOGRMTPRB)

Indicates which problem notifications automatically create problem records in

the service provider problem log. You can allow service requests (*SRVRQS), alerts (*ALERT), or both (*ALL) to automatically log problems when received.

Send PTFs automatically (SNDPTFAUTO)

Allows you to specify whether or not you want released PTFs (PTFs in save files that are not held) to be distributed automatically in response to a service request or PTF order. This attribute helps you to automate some of the problem management operations necessary to maintain a large number of systems.

Note: PTFs cannot be distributed automatically in response to an alert notification.

Message queue (MSGQ)

Indicates where you want to place messages indicating the receipt and processing of a problem notification or a PTF order. If you choose to use a message queue other than QSYSOPR, you should create the message queue before you specify it on the Change Service Provider Attributes prompt display.

Maximum distribution (MAXDST)

Indicates the maximum number of KBs that can be transmitted to a service requester in response to a PTF order or service request.

Hold PTFs received (HLDPTF)

Indicates whether or not you want to place new PTFs you have received from IBM service support (*IBMSRV) or another OS/400 service provider on hold. When you place a PTF on hold, it is not distributed to other systems, even if the Send PTFs automatically attribute is set to *YES. This allows you to test new PTFs to be sure they correct the reported problem before you distribute them to your service requesters.

Cover letter language (CVRLTRLNG)

Indicates the language of the cover letter to send. You can press F4 to see a list of languages.

Forward requests automatically (FWDRQSAUTO)

Indicates whether the requests should be automatically sent to the next service provider.

The service provider attributes take effect immediately and can be changed at any time. To view the service provider attributes, use the Display Service Provider Attributes (DSPSRVPVDA) command.

Setting up the list of service requester servers

The following procedure explains how to add a service requester entry to the list of service requesters that are to be entitled to problem management and change management support from your server. For an example of creating the list of service requesters, see "Setting up the list of service requesters" on page 244.

1. Type the Work with Service Requesters (WRKSRVRQS) command on any command line and press the Enter key.

```

                                Work with Service Requesters
                                System:  HQSYS

Position to . . . . . _____ Control point
Network ID . . . . . _____

Type options, press Enter.
  1=Add  2=Change  3=Copy  4=Remove  5=Display

Opt      Control
          Point      Network ID      Description
  1      CHICAGO1      CUSTNET1      Service Supplies System_____
  -      ACCTSYS       CUSTNET2      Accounting System in Peoria

                                Bottom
F3=Exit  F5=Refresh  F12=Cancel  F13=Configure Distribution Services

```

The Work with Service Requesters display is shown. From this display you can add, change, copy, delete, and display a service requester entry from the list of service requesters entitled to service from your system.

A directory entry and a *SNADS distribution queue need to be manually created if you chose to use *SNADS for receiving PTFs. You must have security administrator (*SECADM) authority to create the directory entry. For more information about directory entries, see Appendix B, "Example setup and configuration of an OS/400 network using SNADS" on page 235.

2. Type a 1 in the *Opt* column followed by the control point name, network ID, and a unique description of the first system you want to add to your list of service requesters.
3. Type the necessary information about the system you are adding to the list of service requesters (only the highlighted fields are required), and press the Enter key. The following example entry will correspond to the service requester CUSTNET1/CHICAGO1 in the network shown in Figure 2 on page 9.

```

                                Add Service Requester

Control point . . . . . : CHICAGO1
Network ID . . . . . : CUSTNET1
Description . . . . . : Service Supplies System

Type information, press Enter.

Contact . . . . . : Sam Jones
Contact telephone numbers:
  Primary . . . . . : 1-312-555-1234
  Alternative . . . . . : 18005551234
Mailing address:
  Street address . . . . . : 1234 Any Street
                                _____
                                _____
  City/State . . . . . : Chicago, IL.
  Country or Region . . . . . : USA
  Zip code . . . . . : 60609
  Machine type . . . . . : 9406
  Serial number . . . . . : 10-12345

                                More...

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel

```

Add Service Requester		
Control point	:	CHICAGO1
Network ID	:	CUSTNET1
Description	:	Service Supplies System
Type information, press Enter.		
Distribution queue type	*SVDS	*SVDS, *SNADS
Service attributes:		
Log remote problems	*SRVPVDA	*SRVPVDA, *NONE, *SRVRQS, *ALERT, *ALL
Send PTFs automatically	*SRVPVDA	*SRVPVDA, *NO, *YES
Message queue	*SRVPVDA	Name, *SRVPVDA
Library		Name, *LIBL, *CURLIB
Maximum distribution	*SRVPVDA	1-5000, *SRVPVDA, *NOMAX
Cover letter language	*SRVPVDA	*SRVPVDA, F4 for List
Forward requests automatically	*SRVPVDA	*SRVPVDA, *NO, *YES
		Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel		

When you type the information on the Add Service Requester display, be sure the machine type and serial number **exactly** match that of the service requester; otherwise, the system will not be entitled to service from your service provider. To find the serial number for an OS/400 service requester, have someone at the service requester use the following command:

```
DSPSYSVAL QSRLNBR
```

If you want to change the service provider attributes that affect the way you provide service for a specific service requester you are adding, page down (roll up) the Add Service Requester display and type the values you want to change in the appropriate fields. If you do not want to change the way service is provided for an individual service requester, you can leave the *SRVPVDA default value in these fields. This indicates that the service provider attributes you set with the Change Service Provider Attributes (CHGSRVPVDA) command are in effect for this service requester.

To specify that the requests received by a service requester be automatically forwarded to the next service provider, specify *YES for the *Forward requests automatically* field when adding a service requester.

Notes:

1. If the Managed System Services program is installed on a service requester, you must use the Work with Service Requesters (WRKSRVRQS) command to indicate that the program is to be used. Change the service requester entry to specify *SVDS instead of *SNADS for the *Distribution queue type* field.
2. A *SVDS distribution queue is needed to distribute objects to remote servers. To distribute objects to non-iSeries servers, you need to add a distribution queue for that system using the Add Distribution Queue (ADDDSTQ) command. These distribution queues are created automatically for iSeries servers when *SVDS is specified for the *Distribution queue type* field on the Add Service Requester display or on the Change Service Requester display.

The service requester you added should be shown in the list. If you have more servers to add, you can use option 3 (Copy) on the Work with Service Requesters display and change only the information that is different for each service requester.

Note: Add only the service requester that you want entitled to service from this server. Follow the setup procedures for each server you want designated as a service provider.

If you need additional controls of security-sensitive objects, you can use the QALWOBJRST system value. This value is used to control whether objects with security-sensitive attributes can be restored. When the operating system is installed, the QALWOBJRST value is ignored and the server functions as if the value is set to *ALL. If the QALWOBJRST value is not *ALL, the security officer may need to change it to *ALL before upgrading to a new release, installing new licensed programs, doing a system recovery, or loading PTFs. You can also specify *ALWPTF. This value enables security-sensitive objects to be restored during a PTF install. For more details about this system value, see the System Value topic in the Information Center.

Setting up product support

The current release of the OS/400 licensed program and System Manager allow an OS/400 service provider to provide product support to service requesters running the current release of the operating system and service requesters running a previous release of the operating system. For a better understanding of release-to-release support, see Table 2 on page 5.

Note: In order to support service requesters, the service provider must be running at the same or higher release of the operating system and System Manager as are the service requesters. It must not be running at a previous release.

When you set up product support on a service provider, System Manager sets up a database of PTF information corresponding to save files containing PTFs and cover letters which you have stored on the system. The database is used to search for PTFs which correct problems similar to the problem with which you are working. If the PTFs found in the search are available on your server and are ready to distribute, they can be sent manually or automatically when a service request or PTF order is received. If you do not want PTFs distributed automatically, you can set the send PTFs automatically (SNDPTFAUTO) service provider attribute to *NO to handle these requests with the intervention of an operator. Regardless of the type of distribution you choose, you need to decide which products, product options, and language features you want to support with each service provider in the network.

Working with supported products (WRKSPTPRD command)

The following section explains the use of the Work with Supported Products (WRKSPTPRD) command and guides you through the displays presented to have the system set up your PTF database. For an example of setting up product support on the service provider, see “Setting up a server as a service provider” on page 14. This procedure is part of step 8, in Figure 1 on page 7.

The WRKSPTPRD command allows the operator at the service provider to specify the list of products, product options, and language features for which service is provided. This includes applications that have been packaged using System Manager. You do not have to have a product or product option installed at the service provider in order to provide support. If the product option and base are different releases, then you will have to install the product and national language versions. To support applications that have been packaged using System Manager

without installing the application on your system, only the product definition is needed. See “Step 1: Creating a product definition” on page 49 for information about creating a product definition.

To speed up the process of adding product support, it is best to use the WRKSPTPRD command before PTF save files accumulate in library QGPL on the service provider.

When you upgrade your service provider to the current release (or modification) of the operating system and System Manager, use the WRKSPTPRD command to add product support for this release. The product support information for previous releases is saved when you install a system upgrade. To update the product support information for a previous release, use the release level (RLS) parameter. The default value for this parameter is *CURRENT. You can print or display all of the product support information for supported releases by specifying *PRINT for the output (OUTPUT) parameter and *ALL for the release level.

Use the following procedure to set up product support on your OS/400 service provider.

1. Type the WRKSPTPRD command on any command line and press the Enter key. The Work with Supported Products display is shown with a list of the products you supported for the current release level.

If you want to add or change product support information for a different release, type WRKSPTPRD and press F4 (Prompt). Specify VxRxMx for the release level parameter, where V stands for version, R stands for release, and M stands for the modification level of the product.

```

Work with Supported Products                                System:  HQSYS
Release:  *CURRENT
Type options, press Enter.
  4=Remove support  5=Display  6=Print
  8=Work with supported language features

  Opt  Product  Product  Description
  ---  -
  -    XXXXXXX  *BASE   Operating System/400
  -    INFOAS4  *BASE   Information APARs
  -    INFODSL  *BASE   Information APARs
  -    XXXXXXX  *BASE   SystemView System Manager/400

Command
====>
F3=Exit  F5=Refresh  F6=Print list  F12=Cancel
F14=Support additional products  F17=Position to  F19=Display trademarks
  
```

2. Press F14 (Support additional products) to add products to the list of products you currently support. A list of all the products and product options for which you can add product support is displayed.
3. Type a 1 (Support) in the *Opt* field of every product and product option you want to support with this service provider and press the Enter key.

Note: If you need to add support for programming requests for price quotations (PRPQs) or for licensed program offerings for the *current*

release of the OS/400 licensed program, you must have them installed on the service provider. PRPQs and licensed program offerings for the previous release do not have to be installed on the service provider to be supported.

4. Press the Enter key again to return to the Work with Supported Products display. Press F5 (Refresh) to show the products for which you added support. Now you can add language feature support if you support systems that are using languages other than the language you use on the service provider.
5. Type an 8 (Work with supported language features) in the option column of each product and product option for which you want to add language feature support, and press the Enter key. The Work with Supported Language Features display is shown with the identifier of the product with which you are working shown at the top of the display.

```

Work with Supported Language Features
System:  HQSYS
Product:  5722SS1   *BASE   Operating System/400
Release:  V5R2M0

Type options, press Enter.
4=Remove support

  Opt   Language   Description
  --   -
  -     2924      English

Command
====>
F3=Exit  F5=Refresh  F12=Cancel  F14=Support additional language features

Bottom

```

Notes:

- a. If this is the first time you have used the WRKSPTPRD command, the language feature shown is either the primary language feature for the product if the product is installed, or the primary language feature for Operating System/400 if the product is not installed on the service provider.
 - b. If the product or product option selected does not have language features associated with it, no entries appear on either the Work with Supported Language Features display or the Support Additional Language Features display.
6. Press F14 (Support additional language features). The Support Additional Language Features display is shown with a list of all language features not currently supported for the product or product option listed at the top of the display.
 7. Type a 1 (Support) in the *Opt* column of each language feature you want to add to the product support for the product or product option listed at the top of the display. Press the Enter key.
 8. Press the Enter key again to return to the Work with Supported Language Features display. Press F5 (Refresh) to include the list of language features for which you added support.

9. Press the Enter key again to either show the Work with Supported Products display or to add language feature support for the next product or product option you selected from the Work with Supported Products display.

From the Work with Supported Products display, you can also display, print, or remove all the support information for a product or product option by typing the option numbers in the *Opt* column next to that product or product option.

Removing support for a product or product option removes the current support information from the PTF database on the system and returns the product or product option to the list of additional products available to support.

If you remove product support for a programming request for price quotation (PRPQ) and the PRPQ is for the current release of the operating system, the product support information is deleted if the PRPQ is not currently installed on the service provider. If this is the case, you must have the PRPQ installed to restore product support. If the PRPQ is for a previous release of the operating system, it is returned to the list of additional products available to support. These same restrictions apply to licensed program offerings.

When you have supported all the products and product options for which you intend to provide service, the service provider automatically takes care of the remaining details.

Now when you receive PTFs for supported products over the electronic customer support service link, the information about those PTFs will immediately be recorded in your PTF database. To work with PTF information and use this information to help manage and maintain your network, see Chapter 6, “Using the program temporary fix (PTF) functions”.

Starting System Manager

When you have completed these steps and the service provider has System Manager installed, you need to run the Start System Manager (STRSYSMGR) command. This corresponds to step 9 in Figure 1 on page 7. The STRSYSMGR command needs to be run only once after the licensed program is installed. The next time the system starts, the System Manager jobs are started automatically. The start and end commands (STRSYSMGR and ENDSYSMGR) can be used to control System Manager operations.

The following jobs are started in the QSYSWRK subsystem to perform system management functions when the Start System Manager (STRSYSMGR) command is run.

QECS Receives PTF requests, service requests, and test requests from service requesters.

QNSCRMON

Processes submitted change requests. This job also starts activities based on start time and conditioning criteria.

QNSSDSTR

Performs initialization and cleanup steps required before running system management functions.

This job only runs for a few seconds or minutes and then this job ends.

Autostart job QNSAJE starts the jobs listed above when the QSYSWRK subsystem starts. If you do not want these jobs to start automatically, remove the QNSAJE entry from the subsystem description by using the Remove Autostart Job Entry (RMVAJE) command.

Starting Manager Services

The Start Manager Services (STRMGRSRV) command starts the services needed at the central site system. The STRMGRSRV command is not included as part of the System Manager licensed program but this step should be included to start the services you need. Specify the service or services that you need to start. The possible values are:

***ALL** Starts all manager services

***RMTCMD**

Allows you to send remote commands to managed systems.

***TOPOLOGY**

Allows you to gather information about systems or nodes in the network.

For more information about this command, see the *Managed System Services for AS/400 Use* book.

Setting up a server as a service requester and a managed system

To set up the server to work efficiently, you need to install Managed System Services on the service requesters or managed systems in the network. See the *Managed System Services for AS/400 Use* book for specific information about installing and using Managed System Services functions. This procedure corresponds to step 10, in Figure 1 on page 7.

Installing Managed System Services on a service requester

Install Managed System Services on all service requesters or managed systems where you want to use additional functions provided by Managed System Services.

To install the Managed System Services program, follow these steps:

1. Type `go licpgm` on any command line and press the Enter key. The Work with Licensed Programs menu is shown.
2. Select option 11 (Install licensed programs) and press the Enter key. The Install Licensed Programs display is shown.
3. Type a 1 (Install) in the *Opt* column next to the licensed program entry, 5722MG1 Managed System Services.

Type `go svms` from any command line to display the Managed System Services menu. Select option 1 (Configure Managed System Services) to display the configuration options. Using option 3 (Work with service providers) on the Configure Managed System Services menu you can set up the systems in the network.

Starting Managed System Services

System Manager and Managed System Services licensed programs work together to manage changes, problems, and operations. System Manager enables an iSeries server to be managed from a central site system.

With System Manager, you can control the operations or activities on the remote or managed servers in your network. System Manager allows you to control the scheduling and conditioning of activities from the central site system. To start System Manager on the service provider, use the Start System Manager (STRSYSMGR) command. System Manager needs to be installed only on the central site system.

Managed System Services needs to be installed on every system in the network, whether it is a managed or central site system. Managed System Services handles all functions required by a managed system in a network, and it generally runs unattended. To start Managed System Services on the service requesters, use the Start Managed System Services (STRMGDSYS) command. This step corresponds to step 10 in Figure 1 on page 7.

Setting up a list of service providers

This procedure must be done on all the service requesters that you want to be entitled to service from one or more OS/400 service providers. If a system is entitled to service from more than one service provider, the procedure must be repeated for each service provider individually. One reason you might set up your network this way is because you want different service providers to provide service for different products. For an example of setting up a list of service providers, see “Setting up the list of service providers” on page 250. This procedure corresponds to step 11, in Figure 1 on page 7.

1. Type the Work with Service Providers (WRKSRVPVD) command on any command line and press the Enter key. The Work with Service Providers display is shown.

Work with Service Providers System: CHICAGO1

Position to _____ Control point
 Network ID _____

Type options, press Enter.
 1=Add 2=Change 3=Copy 4=Remove 5=Display

Opt	Control Point	Network ID	Description
-	*IBMSRV	_____	IBM Service Support
-	HQSYS	CUSTNET	Corporate Headquarters

Bottom

F3=Exit F5=Refresh F12=Cancel F13=Configure distribution services
 F22=Change IBM service route

From this display you can add, change, or copy service provider entries. A service provider entry indicates that this system may send service requests and PTF orders to that service provider. If you chose to use *SNADS for receiving PTFs, you need to create a directory entry and a *SNADS distribution queue. You must have security administrator (*SECADM) authority to create the directory entry. For more information about directory entries, see “Creating

directory entries” on page 235. For more information about SNADS, see Appendix B, “Example setup and configuration of an OS/400 network using SNADS” on page 235.

The IBM service support (*IBMSRV) service provider entry is shipped with the system, and its contents are initially blank. The service representative should add the necessary telephone numbers when the system is installed.

You cannot add, remove, or copy this service provider entry; however, you can change the route to IBM service support (*IBMSRV) using the function key provided. When the route to IBM service support is by way of an OS/400 service provider, the *IBMSRV entry is not shown. One reason for doing this is that a business with a large network may want to minimize the number of lines going to IBM service support (*IBMSRV).

2. Type a 1 in the *Opt* column followed by the control point name, network ID, and a brief description of a service provider and press the Enter key.

```

                                Add Service Provider
                                System:  SYSTEM
Control point . . . . . : HQSYS
Network ID . . . . . : CUSTNET
Description . . . . . : Corporate Headquarters

Type information, press Enter.

Contact . . . . .
Contact telephone numbers:
  Primary . . . . .
  Alternative . . . . .
Mailing address:
  Street address . . . . .

  City/State . . . . .
  Country or Region. . . . .
  Zip code . . . . .

                                Bottom

F3=Exit   F5=Refresh   F12=Cancel
  
```

3. Type the necessary information about the service provider in the fields provided and press the Enter key. The operator at the service provider you are adding can assist you if you are unsure of what to type in any field.

Note: The system automatically creates an *SVDS distribution queue only if Managed System Services is installed.

4. Repeat these steps to add more service providers to the list, or select option 3 (Copy) to use your first service provider entry as a basis for another entry. This can save time if there are many systems in your network acting as service providers.
5. SNA distribution services (SNADS) needs to be started on the Service Requester to use object distribution or *SVDS distribution. A job called QESTP is started to receive the PTFs automatically using object distribution.
6. If the Managed System Services program is installed, use the Start Managed System Services (STRMGDSYS) command to start jobs such as receiving objects, running programs, running commands, and applying program temporary fixes (PTFs). You do not need to repeat this command for each service provider added to the network. This step only needs to be done once to use the *SVDS distribution function.

Once you have added at least one OS/400 service provider to your list, you can change the route to IBM service support (*IBMSRV). You need to do this if you intend to send problem reports and PTF orders only to OS/400 service providers.

On the Work with Service Providers display, do the following:

1. Press F22 (Change IBM service route) to change the route to IBM service support (*IBMSRV).
2. Type the control point name and the network ID of the OS/400 service provider you want to use as the new route to IBM service support (*IBMSRV) and press the Enter key.

The entry for IBM service support (*IBMSRV) is no longer shown in the list of service providers.

Resolving problems

After you have completed your list of service providers, you may want to create a user profile to allow the network operator at the service provider to sign on to your system and to run remote problem analysis procedures. For an example of creating the user profile for running remote problem analysis, see “Creating a user profile for remote problem analysis” on page 252. This procedure corresponds to step 12, in Figure 1 on page 7.

The network operator must have the authority to use the Work with Problem (WRKPRB) command, the Analyze Problem (ANZPRB) command, and the Print Error Log (PRTERRLOG) command. The network operator may also need to be authorized to objects on the service requester that may be used during problem analysis. Some of the objects that may be used during problem analysis include line, controller, and device descriptions that are related to the problem.

When you create the user profile for running remote problem analysis, you can set it up with a user class of *SYSOPR. This is adequate to run remote problem analysis once you grant the authority to the commands and objects. You may want to assign a user profile with a higher user class than *SYSOPR; however, it is recommended that you do not assign the *SECOFR user class.

When performing remote problem analysis, the network operator must be authorized to the APPC device being used for the incoming pass-through session on the service requester. This device on the service requester is used to accommodate the pass-through session from the service provider.

You may also need to change two system values which correspond to signing on to a remote server. System value QAUTOVRT (automatic virtual device configuration) allows you to limit the number of concurrent remote sessions. System value QRMTSIGN (remote sign-on verification) lets you indicate what type of security you want to have for remote sign-on. You must have *SECADM and *ALLOBJ authority to change the QRMTSIGN system value. If you do not have this authority, contact the security officer for your system and have them set the value for QRMTSIGN.

The following procedure provides parameters for a basic user profile that can be used by all the service provider operators who need access to the service requester for problem management tasks.

1. Type the Create User Profile (CRTUSRPRF) command on any command line. The Create User Profile display is shown.

2. Type the user ID and password you want the operator at the service provider to use to sign on to your system.
3. Type *SYSOPR in the *User class* field and press the Enter key.

Provide the user ID and the password to the network operator at the service provider.

To grant the authority for the commands necessary for running remote problem analysis, use the Grant Object Authority (GRTOBJAUT) command. Type the following commands and parameters on any command line, specifying the user ID from the profile you just created:

- Type GRTOBJAUT OBJ(WRKPRB) OBJTYPE(*CMD) USER(USERID), and press the Enter key.
- Type GRTOBJAUT OBJ(ANZPRB) OBJTYPE(*CMD) USER(USERID), and press the Enter key.
- Type GRTOBJAUT OBJ(PRTERLOG) OBJTYPE(*CMD) USER(USERID), and press the Enter key.

After you create the user profile and grant the authority to the necessary commands, your final step is to change the two system values QAUTOVRT and QRMTSIGN:

- Type CHGSYSVAL SYSVAL(QAUTOVRT) VALUE(XX), where XX is a number greater than 0 and press the Enter key.
- Type CHGSYSVAL SYSVAL(QRMTSIGN) VALUE(*VERIFY) and press the Enter key.

Note: The *VERIFY value for the remote sign-on control system value indicates the service requester must verify that the user signing on has access to the system.

The setup procedures for the service requester are now complete.

Verifying the network setup

When the procedures for setting up the service providers and service requesters in your network are complete, you should have the operators at the service requesters use the Send Service Request (SNDSRVRQS) command with the *TEST parameter to verify the systems are set up correctly. Have the operator at each service requester perform the following procedure and look for messages displayed to determine the results. This procedure corresponds to step 13, in Figure 1 on page 7.

1. Type SNDSRVRQS on a command line and press F4 (Prompt).
2. Type *TEST in the *Action* field and press the function key provided to show the additional parameters for the destination of the request.
3. Type *IBMSRV or type *SELECT in the *Remote control point* field to choose the destination of the test request.
4. Press the Enter key to send the test service request.

Status messages are shown on the bottom of the service requester display that indicate whether the request is being sent or if there is a problem.

A test request can always be sent to IBM service support by specifying *IBMSRV in the *Control point name* field. If a service requester has changed the route to IBM service support (*IBMSRV), the OS/400 service provider indicated as the route to IBM service support (*IBMSRV) forwards the test request to the next level service

provider until the IBM service support system is reached. Status messages are shown to indicate whether the test request was successfully forwarded and whether it reached the specified destination.

When the test request arrives at the destination, a check is done to verify the service requester is entitled to service from that service provider. If the destination is an OS/400 service provider and the service requester is entitled to service, a message indicating that the test request was successful is sent. If the destination is IBM service support (*IBMSRV), a message is also sent, and a service representative contacts the service requester to verify the request was received.

The test request may not always be successful. For example, if a communications line is down or a service requester is not properly added to the list of supported systems at the service provider, a test request will fail. In all cases, a message is shown on the display at the service requester that will provide an indication of what caused the request to fail. If there is an equipment problem, the operator at the service requester should check the job log for messages relating to the test service request.

No manual intervention is needed at the service provider to process a successful test service request. Some intervention may be needed if the test service request fails because of mismatched information between the service requester and service provider, or if the entitlement checking at the service provider fails.

When the service provider has received or forwarded test service requests successfully from each service requester in the network, verification of the setup procedures is complete. Once you have verified the setup procedures, you can begin using the problem management and change management functions provided by System Manager.

Note: The Send Service Request (SNDSRVRQS) command with the *TEST parameter does **not** verify the SNADS configurations.

The following examples are provided to assist you with setting up your network. These examples provide general information; however, they may not apply specifically to your network. For a more complete example see Appendix B, "Example setup and configuration of an OS/400 network using SNADS" on page 235.

Example networks

This section shows example networks and some of the commands and parameters you will use to set up your network to use System Manager. These are not the only possibilities for setting up a network. For the sake of simplicity, the detailed information about APPN communications and SNADS setup has been left out and the second example is not as detailed as the first. For a complete example of setting up a network of iSeries servers for both communications and System Manager, see Appendix B, "Example setup and configuration of an OS/400 network using SNADS" on page 235.

iSeries example network

Figure 3 on page 32 shows two networks of iSeries servers where the highest level of problem management and change management support is provided by an iSeries server. Each service provider may provide support for a different set of applications. For example, SYSB might be an IBM business partner. Each service

provider is an alert primary focal point and has a direct connection to IBM service support (*IBMSRV) and IBM Marketing support (IBMLINK). Service provider SYSA provides support for two service requesters, SYSA1 and SYSA2. Service provider SYSB provides support for two service requesters, SYSB1 and SYSB2.

Note: For the example, assume that SNA distribution services (SNADS) has been set up before System Manager was installed on SYSA and SYSB.

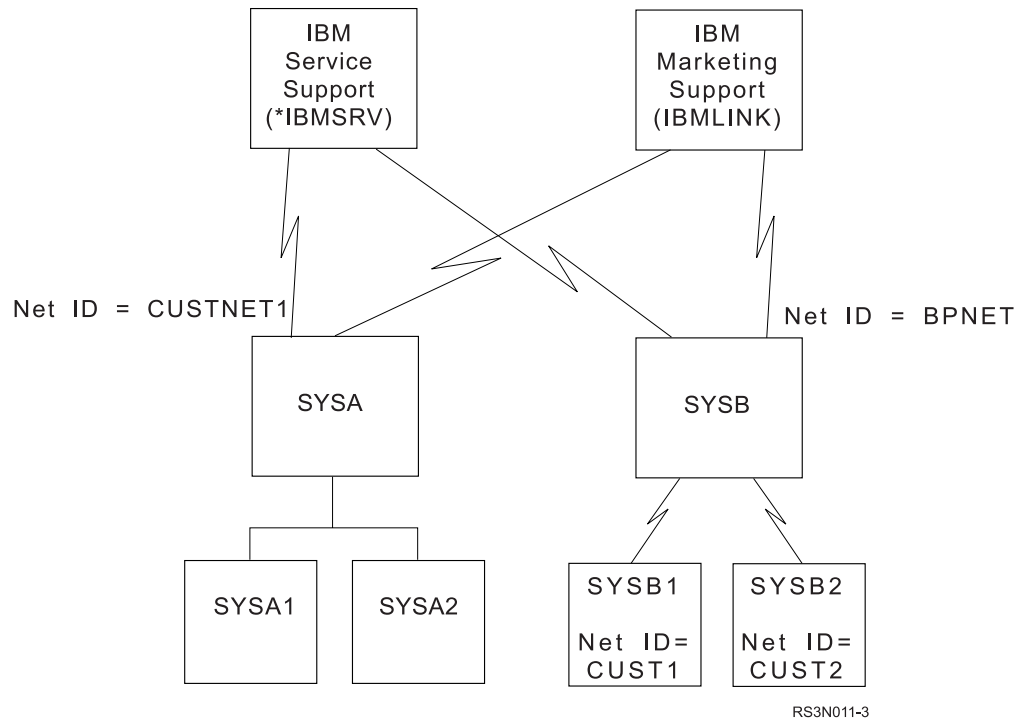


Figure 3. Two iSeries Networks

With System Manager installed, SYSA and SYSB can provide service and support for their service requesters.

Setting up System Manager

Service providers SYSA and SYSB want to be able to receive alerts from their service requesters. Using the Change Network Attributes (CHGNETA) command, the alert support network attributes are set to the values shown in Figure 4.

```

NODETYPE = *NETNODE
NETSERVER = *NONE
ALRSTATUS = *ON
ALRPRIFP = *YES
ALRDFTFP = *NO
ALRLOGSTS = *ALL
ALRCTLD = *NONE
  
```

Figure 4. Service Providers Network Attributes for Alert Support

Once alert support is set up and the System Manager program is installed, the operators at the service providers need to grant authority to the service provider commands. When the commands are available to use, the operators at the service providers need to set the service provider attributes that control the creation of problem records for remote problems and the PTF distribution functions. This is

accomplished by using the Change Service Provider Attributes (CHGSRVPVDA) command. The list in Figure 5 on page 33 shows the values for the service provider attributes chosen for SYSA and SYSB in the example:

Next, a list of the service requesters that are entitled to service must be created

```
LOGRMTPRB = *ALL      (Log remote problems)
SNDPTFAUTO = *YES     (Send PTFs automatically)
MSGQ       = NETOPR   (Service provider message queue)
LIB        = QGPL     (Message queue library)
MAXDST     = 500     (Maximum distribution)
HLDPTF     = *YES     (Hold PTFs received)
CVRLRLNG   = *YES     (Cover letter language)
FWRDQSAUTO = *YES     (Forward Change Requests Automatically)
```

Figure 5. Service Provider Attributes

using the Work with Service Requesters (WRKSRVRQS) command. In our example, the operator at SYSA uses the WRKSRVRQS command to add SYSA1 and SYSA2 to the list of systems entitled to service from SYSA. The operator at SYSB does the same for SYSB1 and SYSB2.

Finally, the service providers need to indicate the products (licensed programs), product options, and associated language features for which they intend to provide service. Using the Work with Supported Products (WRKSPTPRD) command, the operators at the service providers can view the products and product options currently supported, and use the function keys provided to add support for more products, product options and associated language features.

When the addition of product support is complete and the setup on the service providers is complete, the System Manager functions are ready to use.

Meanwhile, at the service requester sites, alert support is also being set up. Service requesters SYSA1 and SYSA2 are set up as end nodes. In order to create alerts and send them to SYSA, they need to use the Change Network Attributes (CHGNETA) command to set the following network attributes to the values shown in Figure 6. SYSB1 and SYSB2 have their network attributes set to these values with the

```
NODETYPE = *ENDNODE
NETSERVER = CUSTNET1/SYSA
ALRSTATUS = *UNATTEND
ALRPRIFFP = *NO
ALRDFTFP  = *NO
ALRLOGSTS = *LOCAL
ALRCTLD   = *NONE
```

Figure 6. SYSA1 and SYSA2 Network Attributes for Alerts

exception of the network server attribute (NETSERVER), which is set to BPNET/SYSB, and the alert controller description, which is set to a SYSBCTL.

After alert support is set up on the service requesters, each service requester must define a list of one or more service providers. Using the Work with Service Providers (WRKSRVPVD) command, the operator at SYSA1 adds system CUSTNET1 / SYSA to the list of systems that will provide service for SYSA1. The operator then uses the function key provided to change the route to IBM service support (*IBMSRV) to a route through SYSA. This eliminates the need for a separate line to IBM service support (*IBMSRV) and decreases line and communications costs.

SYSA2 follows the same procedure for adding SYSA as its service provider and route to IBM service support (*IBMSRV), while the operators at service requesters SYSB1 and SYSB2 use the Work with Service Providers (WRKSRVPVD) command to add SYSB as their service provider and to designate SYSB as their route to IBM service support (*IBMSRV).

Finally, the operator at each of the service requesters must create a user profile and authorize the user to the necessary commands for problem management. This allows the operator at the service provider to run remote problem analysis. When this step is complete, both networks are set up to use System Manager functions for problem and change management.

System/370 and iSeries Example Network

Figure 7 shows a network of iSeries servers where the highest level of problem management and change management support is provided by an iSeries server and a System/370 or System/390 NetView® help desk. The connection to IBM service support (*IBMSRV) is only through SYSA, the iSeries server in the central location.

Note: In the example, assume that SNA distribution services (SNADS) has been set up before the System Manager licensed program was installed on SYSA and SYSB.

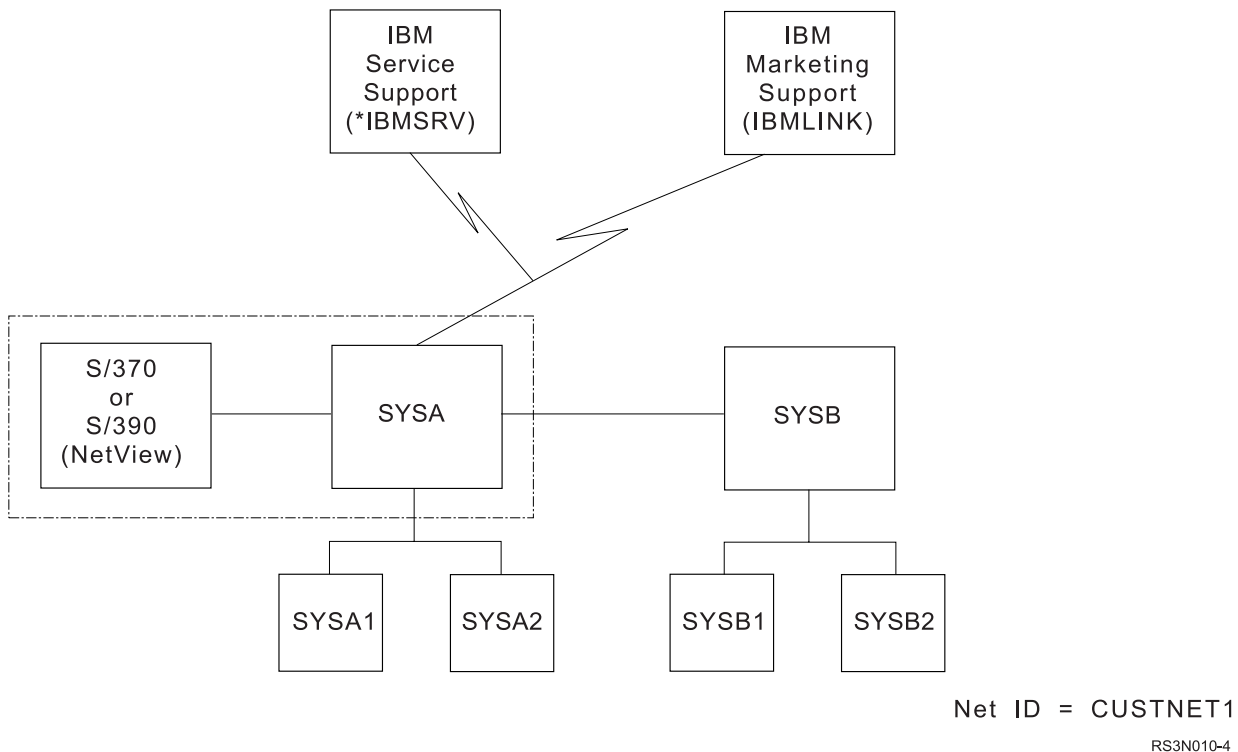


Figure 7. A System/370 or System/390 with an iSeries Network

With System Manager installed, SYSA and SYSB can provide service and support for their service requesters.

Setting up System Manager with Host System/370

Service providers SYSA and SYSB want to receive alerts from their service requesters and possibly forward these to the System/370 or System/390. In the

environment shown in Figure 7 on page 34, SYSA provides service for service requesters SYSA1 and SYSA2. In addition, SYSA provides service for and is the alert primary focal point (ALRPRIFF) for SYSB, also an OS/400 service provider. Using the Change Network Attributes (CHGNETA) command, the network attributes for SYSA are defined as shown in Figure 8.

```
NODETYPE = *NETNODE
NETSERVER = *NONE
ALRSTATUS = *ON
ALRPRIFF = *YES
ALRDFTFP = *NO
ALRLOGSTS = *ALL
ALRCTLD = HOST370
```

Figure 8. Service Providers Network Attributes for Alert Support

The network attributes for SYSB are similar, with the exception of the alert controller description (ALRCTLD), which is set to *NONE. Instead, using the Work with Sphere of Control (WRKSOC) command, SYSA has added SYSB to its sphere of control.

Once alert support is set up and the System Manager licensed program is installed, the setup then proceeds as shown in section “Setting up System Manager” on page 32. The one difference is SYSB, which is both a service provider for SYSB1 and SYSB2 and a service requester for SYSA. For this special case, the operator at SYSB has to perform the setup activities for both a service provider and a service requester.

First, at SYSB, the operator adds SYSB to the list of service providers. The operator uses the Work with Service Providers (WRKSRVPVD) command and the function key provided to indicate SYSA is the route to IBM service support (*IBMSRV). Next, a user profile is created on SYSB to allow the operator at SYSA to run problem analysis remotely on SYSB. Now, the operator at SYSB can begin the procedures for setting up as a service provider.

Meanwhile, the service requesters are being set up as well. The alert network attributes for these systems are the same as those shown in Figure 6 on page 33, with the exception of the network server attribute (NETSERVER). For SYSB1 and SYSB2, the network node server is CUSTNET1/SYSB. The remainder of the setup for SYSA1, SYSA2, SYSB1, and SYSB2 is the same as that of the service requesters in “iSeries example network” on page 31.

Chapter 2. Product packaging concepts

A product is an application program that has been packaged using the System Manager licensed program. The iSeries server provides management functions for software that is identified as a product. To use the management functions for your own software, the software must be identified to the iSeries server as a product. However, before you convert your application into a product that the iSeries server identifies as a product, you must be familiar with some important packaging concepts.

This chapter explains important product development concepts that you need to understand before you package your products with System Manager. Some of the concepts that this chapter discusses include:

- Prepackaging considerations
- National language support
- Dynamic naming
- Packaging with libraries, folders, and directories
- Product development, testing, and usage modes
- Establishing a development environment for directories
- Storing document library objects in auxiliary storage pools
- Working with database files
- Packaging a Follow-on Release
- Repackaging considerations

Understanding these concepts helps you make better decisions when you complete the packaging steps in Chapter 3, “Packaging your products” on page 47.

Prepackaging considerations

Before packaging your product with System Manager, consider the following:

- Marketing and delivery of your product
- Application environment of your product
- Developing the process for your product
- Maintaining your product

Marketing and delivery

Options and Features.

- How do you intend to market your product?
- Which pieces of your product are required, and which are optional?

See “Step 1: Creating a product definition” on page 49.

User-Based Priced Products.

- Do you plan to base the purchase price of your product on the number of its users?

See Chapter 4, “Packaging products for licensing” on page 93.

Translation.

- Do you plan to make your product available in more than one national language version (NLV)?

If the language of the systems where your product is installed is a mixture of both English and Spanish, then consider providing both an English and a Spanish version of the product.

See “National language support” on page 41.

Product Ownership.

- Do you plan on using your software internally only?
- Do you plan to distribute the software for use within a single company or for use in many companies?
- Do other products share libraries with this product?
If so, who develops these products? (Your development organization or a different development organization?)

See “Step 1: Creating a product definition” on page 49 for information on registration identifiers.

Corequisite products.

- Does your product require the installation of another product on the system?

See “Product packaging restrictions” on page 48.

Security.

- What user profile owns the objects once the product is installed on the system?
- Is special authority required to install your product?
- Will the user profile exist on the system before you install the product?

See the *iSeries Security Reference* book.

Instructions.

- Do you need to provide installation instructions for your product?
- Are any special user profiles required?

See Chapter 5, “Product installation, distribution, and management”.

Redistribution.

- If the customer is responsible for distributing your product to others, do you need to supply instructions about how to perform the operations?

See Chapter 5, “Product installation, distribution, and management”.

You need to compile the object at the correct release level. This ensures that products contain objects compatible with the release level to which they are being restored.

The application environment

Some items to consider when setting up an application environment are:

- Test versus production
- Multilingual support

- Data file/database replacement
- Library lists
- Install new release versus replace current release
- OS/400 release level

Test versus Production (Multiple Releases).

- Do you need to develop, package, and test the product in test mode while a different release of the same product is in production mode on the same system?

See “Multiple releases and repackaging considerations” on page 42.

Multilingual.

- Do two or more different national language versions (NLV) of your product exist on the system at the same time?

See “National language support” on page 41.

Data File/Database Replacement.

- When you install a new release of your product, do you need to handle any files that currently exist on the system for your application?
See “Exit program functions for restoring a product” on page 75.
- Are there special considerations for packaging database files?
See “Working with database files” on page 46.

Library Lists.

- If you have an existing application, is more than one library used?
- If you do not have an existing application, will you use more than one library?

See “Packaging products that have more than one library” on page 55 and “National language support” on page 41.

Install New Release versus Replace Current Release.

- If your product currently exists on the system, do you need to keep the existing files or libraries for any future installations of next product version?
- Do any of the files or libraries contain customer data that the new release of the product needs to use?

See “Product exit program interface” on page 74.

Maintenance

Program Temporary Fixes (PTFs).

- Will you supply PTFs for your product?
- Do you have a process in place for servicing objects in case that your product changes?
- Do you need to create a plan or process for servicing your product?

See Chapter 7, “Servicing your products using System Manager” on page 141.

Instructions to your customer.

- You must supply your customers with instructions on how to report any problems encountered with your product.

- What information do you want them to supply?
- Do you want them to use the Analyze Problem (ANZPRB) command?
- If you supply PTFs for your product, you also need to supply instructions for loading and applying the PTFs.

See Chapter 7, “Servicing your products using System Manager”.

Release Level.

- Is more than one release of your product available at the same time?
- Do you create or supply PTFs for all available releases?

See “Multiple releases and repackaging considerations” on page 42 and Chapter 7, “Servicing your products using System Manager”.

OS/400 Release Level.

- Install the products that are created with System Manager.

Problem Management.

- Will you service your product by using electronic customer support?
- Will you provide remote problem analysis?
- Will you support using electronic customer support that retrieves PTFs for your product from an OS/400 service provider?

See “Using the problem log commands” on page 157.

Development Process

CL Programs.

- Commands that package products contain create commands and delete commands, without change commands. The create commands may contain a large amount of information. Use CL programs or batch job streams for these commands instead of using the commands interactively.
- If you are using directories with exit programs, do you need to convert your exit program from CL to another programming language?

Development Libraries.

- Do you plan to develop your product in the same libraries or folders in which you plan to install the product?
- Do you have a product with directories installed on your system already?

See “Packaging libraries, folders, and directories” on page 43.

Avoiding Name Conflicts.

- Do you have more than one product or option that exist in the same library?
- If your products are multilingual, do you have more than one product that uses the same library when it is installed as a secondary language?
- Can your product handle having a customer specify the library and folder names for your product at installation time?
- Does your program allow for redirection to avoid naming collision?
- To avoid collision, do your directory path names contain home directories?

See “Dynamic naming” on page 41.

WebSphere® Development Studio for iSeries™ Licensed Program.

- This licensed program, 5722-WDS, provides application development organizations with the ability to manage application objects and source code effectively throughout the life of the application.

The WebSphere Development Studio product allows a group of developers to create, manage, organize, and build multiple versions of their application. At the same time, this licensed program maintains the integrity of the user's application by not allowing one developer to overwrite another developer's source changes.

You can use this product to assist you in managing the development of the application. Once you have the application for use in the production environment, use the System Manager program to prepare the application for distribution and installation to other systems.

National language support

If you will be translating an option for your product, the option must contain a **language load**. The language load contains all the product objects you will be translating, such as menus and messages. The language load also contains one product load object, and it can also contain exit programs.

When you create the product definition, you define a list of options that will be shipped with the product. The option information requires a list of **language load identifiers**. A language load identifier identifies a language in which the option could become available. To see an example of creating a national language load, see "Understanding national language loads for your product" on page 67. The identifiers are of the format 29xx, where 29xx is a valid IBM NLV. For a list of valid IBM NLVs, see the *Software Installation* book.

Specify the primary library names on the PRILIB parameter and ADLLIB parameter of the Create Product Load (CRTPRDLOD) command. If the primary library names are not the same, the languages for the option will not install correctly. Therefore, when the system creates each language load, you must make sure that the primary library names are the same as the primary library names for the corresponding code load. See "Product development, testing, and usage modes" on page 43 for information about development libraries and primary libraries. A product option can have a maximum of one language load for each NLV supported by the iSeries server.

Dynamic naming

Dynamic naming allows you to install your product into libraries, folders, or directories other than the one that you defined when you packaged the product. This allows multiple releases of the product to be on the system at the same time. If you cannot restore the product load into libraries, folders, and product home directories with names other than those specified on the Create Product Load (CRTPRDLOD) command, the load does not allow dynamic naming. When you create the product definition, this attribute is set for each option. Note that if an option has more than one folder, you can dynamically name the root folder.

To set the dynamic naming attribute, use the product option (PRDOPT) parameter on the Create Product Definition (CRTPRDDFN) command. See "Step 1: Creating a product definition" on page 49 for more detailed information. The dynamic naming attribute controls whether you can specify LIB, LNGLIB, FLR, CODHOMEDIR, and LNGHOMEDIR parameters on the Restore Licensed Program (RSTLICPGM) command.

If dynamic naming is not allowed, you may be unable to install your product. This can occur when another vendor may be using libraries, folders, or product directories that your product also uses. This situation is referred to as collision. Specifying dynamic naming attribute while completing “Step 1: Creating a product definition” on page 49 prevents collision.

Benefits of dynamic naming include the following:

1. Avoiding collision among objects, documents, stream files, directories, and folders with the same name
2. Providing multiple release support
3. Providing directory development support

See “Establishing a development environment for directories” on page 45 for more information.

Dynamic naming restrictions

When you specify the dynamic naming attribute while completing “Step 1: Creating a product definition” on page 49, you must be aware of the following restrictions:

1. All loads for an option must support dynamic naming.
2. All libraries within a load also must support dynamic naming.
3. You can only dynamically name the root folder.
4. You can only dynamically name the home directory.

The following is an example of how the Create Product Definition (CRTPRDDFN) command may look if both options of the sample product allowed dynamic naming:

```
/*
/*****
/* Define the product to the system with the
/* allow-dynamic-naming attribute.
/*****
/

CRTPRDDFN PRDDFN(COMPILERZ/PRDDFN) +
          PRDID(7CE6D11) +
          RLS(VIRIM0) +
          RGSID(*PHONE 0005072534011) +
          MSGF(CMPMSG) +
          PRDOPT((*BASE MSG0001 *ALWDYNNAM) +
                (1 MSG0002 *ALWDYNNAM)) +
          CPYRGTFST(*CURRENT)
```

Note: Remember that you can accomplish dynamic naming by using the Restore Licensed Program (RSTLICPGM) command.

Multiple releases and repackaging considerations

Two or more versions of the product at different release levels can be installed on the server at the same time. When this occurs the product allows multiple releases. If you want to maintain multiple releases, including creating PTFs, you must have the product installed on your server at the release level for which you are providing the PTF. You may need to specify the allow-multiple-releases parameter for your product as well as the dynamic naming attribute so more than one release of the product can be installed on the same server.

Not all products can function properly when more than one release of that product is installed. If your product must have this restriction, then you should not allow multiple releases when you complete “Step 1: Creating a product definition” on page 49. The Restore Licensed Program (RSTLICPGM) command then prevents a second release of the product from being installed unless the second release replaces the release currently installed.

If a product does not support dynamic naming, a second release of the product must use a set of different library, folder, and directory names if multiple releases are to be installed without one replacing the other.

If you change the principal primary library from release to release and do not want multiple releases installed on the iSeries server, your product exit programs should delete the previous release when installing a new release. For information about exit programs, see “Step 3: Packaging products that have exit programs” on page 72. For information about primary libraries, see “Product development, testing, and usage modes”.

This attribute is set when the product definition is created using the ALWMLTRLS parameter on the Create Product Definition (CRTPRDDFN) command.

Packaging libraries, folders, and directories

Your product may contain software that resides in libraries, folders, or directories. Objects associated with multiple product loads can reside in the same library. A product load cannot contain both folders and directories. Folders and documents that reside in folders can only be allocated to a single product load. Directories and stream files that reside in directories can only be allocated to a single product load.

You can package your product by using **development libraries** and **development folders**. System Manager does not provide support for packaging your product by using development directories. Because you cannot establish a development directory, you may use a special procedure to package your directories. See “Establishing a development environment for directories” on page 45. If the product option contains directories, you should allow dynamic naming for the option. If you do not allow dynamic naming, you will have to package the product with a different home directory than the one defined in a prior release. Remember that if you do not allow dynamic naming, the system may not allow you to install your product if your product has a different registration ID installed in the directory. For more information see “Dynamic naming” on page 41.

After the product has been packaged, it can be saved with the Save Licensed Program (SAVLICPGM) command. When it is restored using the Restore Licensed Program (RSTLICPGM) command, it is installed into the libraries, folders, or directories that are specified on the Create Product Load (CRTPRDL0D) command. For more information about the license program commands, see “Step 7: Saving and restoring your product” on page 85.

Product development, testing, and usage modes

The following terms depict the modes of a product:

- Development mode
- Test mode
- Production mode

A product in **development mode** has not yet been released to a customer. It is still being developed. The product exists in development libraries, development folders, or directories.

A product that is in **test mode** has been installed on the system with the Restore Licensed Program (RSTLICPGM) command. However, it is not yet being used as the production version of the product. The product in test mode no longer exists in development libraries or development folders, but exists in product libraries, product folders, or product directories. If the product option allows dynamic naming, the library names, folder names, or product home directory names can be modified if collision occurs when the Restore Licensed Program (RSTLICPGM) command is used. See “Dynamic naming” on page 41 for more information on dynamic naming.

A product in **production mode** is installed on a system and is currently being **used** for its intended purpose. The product in production mode no longer exists in development libraries and development folders, but exists in product libraries and folders, or directories.

Note: Do not confuse these terms with a library type. Libraries can have a type of test (*TEST) or production (*PROD). Development mode, test mode, and production mode can use libraries independent of their type.

Following is an example of when both types of libraries would be used:
Two different releases of the same product are to exist on a system at one time, one release in development mode, and the other release installed and in production mode. The release of the product that is under development exists in the development library, while the release of the product that is in production mode exists in the primary library.

In this example, care should be taken to ensure that objects in a common library would not be used by both releases of the product. For example, if both releases use a file in library QGPL, special handling would be required. This ensures that the release in test mode does not interfere with the data for the release in production mode.

If you apply the concept of principal and additional libraries to the concept of development and primary library, the following combinations of terms arise:

- **Principal development library**
 - The library where the product load object is created.
- **Principal primary library**
 - The library where the product load object will normally be installed.
- **Additional development library**
 - Another library where application objects are created.
- **Additional primary library**
 - Another library where application objects will normally be installed.

Note that the principal development and principal primary libraries comprise the **principal pair** of libraries. The additional libraries are similarly paired.

Similar to primary and development libraries, there is a concept of primary and development folders. Following are additional rules for packaging a product with folders when development folders and primary folders have been specified:

- If a language load has folders, the first development folder must be a subfolder of the first development folder for the code load.
- If a language load has folders, the first primary folder must be a subfolder of the first primary folder for the code load.

See “Example of packaging a product by using folders” on page 57 for an example of using development libraries and development folders.

System Manager does not support development directories. If your product contains directories and multiple copies exist you should read the section that follows.

Establishing a development environment for directories

The Create Product Load (CRTPRDLOD) command does not allow you to create development directories. The following procedure allows you to package a new version of a product load with directories. If your product contains directories and you plan to develop other releases of the product, then you should allow dynamic naming in the initial release of the product. You also do not need to follow this procedure if your product load does not contain directories. Completing the procedure prevents you or your customers from having to rename your product home directories when you restore your license program.

If this is the first release of the product with product directories then specify the dynamic naming attribute. See “Step 1: Creating a product definition” on page 49. If this is a follow-on directory of the product which contains home directories that were defined in a previous release and packaged in the same version, then complete the following steps:

1. Save the product loads of the product option by using the Save License Program (SAVLICPGM) command.
2. Delete the product option by using the Delete Licensed Program (DLTLIPGM) command.
3. Restore the product option by using the Restore Licensed Program (RSTLICPGM) command.

Note: If you have not allowed for dynamic naming, you will not be able to complete this step. See “Dynamic naming” on page 41.

4. Create all product loads by using the product home directory that will be shipped with the product. The system creates the product load with the home directory names in which it will be installed.
5. Create product home directories. Each product home directory that is defined in the product load must be created before it can be packaged. See “Packaging products that have directories” on page 61.

This procedure provides you with the ability to use the same product home directory names across multiple releases. You must specify the dynamic naming attribute during “Step 1: Creating a product definition” on page 49. **If you do not do this procedure, it will not work!**

Storing document library objects in auxiliary storage pools

You can store document library objects (DLOs) in user auxiliary storage pools (ASPs). With this option you can allocate your DLOs in different pools with different recovery strategies that are based on the importance of the DLO.

If you use the SAVLICPGM command to save a product that has DLOs in a user auxiliary storage pool (ASP) and the product that is installed on another system, the DLOs also are installed into the system ASP. To put your DLO into your ASP use the Save DLO (SAVDLO) or Restore DLO (RSTDLO) commands.

Working with database files

Sometimes you may want to enhance your application without changing the format of the database files. If so, consider packaging your database files as a separate product. This allows you to upgrade the application without having to redistribute the database. For more information about how to package a database as a product, see “Step 6: Packaging your product option” on page 80.

Packaging a follow-on release

When developing a new release of your product, you need to create a product definition for the new release and create new product load objects, as well as make any changes or additions to your functional objects. Before packaging the new release, update the object descriptions of all objects for the new release to indicate the new release level using the Change Product Object Description (CHGPRDOBJD) command. See “Packaging products that have directories” on page 61.

Repackaging considerations

Once a product load is packaged successfully, you can repackage it. Specify REPACKAGE(*YES) on the Package Product Option (PKGPRDOPT) command. If an error occurs during the repackaging function, the product load can no longer be saved by the Save Licensed Program (SAVLICPGM) command. This is because it is the packaging function that prepares a load for use by the SAVLICPGM command. Unsuccessful packaging or unsuccessful repackaging prevents the load from being saved by the SAVLICPGM command. Therefore, repackaging an installed product load can cause the product load to become unusable by the SAVLICPGM command until the load is repackaged successfully.

You cannot repackage a load that is installed as a secondary language if it has an additional library. Therefore, to repackage a language load that is installed as a secondary language, either install that load on a system with the appropriate primary language or delete and recreate the objects for that load (including the product load object).

Chapter 3. Packaging your products

Your present application may contain elements that exist in the libraries, folders, and directories. Libraries contain objects that you can manage by using OS/400 library and object commands. To see a listing of the commands that allow you to work with libraries, type GO CMDLIB at the command line. If you want to see a listing of commands that allow you to work with objects, type GO CMDOBJ at the command line.

Folders contain documents that you can manage by using OS/400 folder and document commands. To see a listing of the commands that allow you to work with folders, type GO CMDFLR at the command line. If you want to see a listing of commands that allow you to work with documents, type GO CMDDOC at the command line.

Directories contain stream files that you can manage by using OS/400 directory and link commands. To see a listing of the commands that allow you to work with directories, type GO CMDDIR at the command line. If you want to see a listing of commands that allow you to work with stream files, type GO CMDLNK at the command line.

System Manager provides you with the capability to convert your application into an OS/400 license program (LP). Packaging is the process of converting your application into a product that contains the following set of hierarchical structures:

- **Product**

The product contains all the elements that are distributed with your application. Your product may contain both required and optional functions. Required functions of a product generally include the base option. Functions that are not required generally are shipped in an option other than the base option. To learn more about the product and its option descriptions see, "Step 1: Creating a product definition" on page 49.

- **Product features**

A product feature is a licensed part of your product that contains one or more product options. All the product options within a feature share a common code load ID.

- **Product option**

A product option is a set of functions that is distributed as a unit. All software that is distributed with a product option contains a common option ID. You can divide the product option into language and code functions which are commonly referred to as product loads. A product option always has a single code load and one or more language loads.

- **Product load**

Product loads are smaller structures of a product option. Product loads contain code and language loads which exist in libraries and folders, or directories. Code loads contain the executable code for your software. Language loads allow you to support the parts of your product that need to be translated. Examples of translatable parts of your product include language loads for menus, messages, and screens.

- **Contents of the product load**

Product loads contain software elements that exist in libraries, folders, and directories. To package your application into a product, this chapter focuses on helping you complete the following tasks:

1. Creating a product definition
This allows you to define the product and its options.
2. Creating a product load
This allows you to define the product structures to develop and install your product.
3. Creating a product message file
This allows you to create messages for your product.
4. Creating exit programs
This allows you to pass control to your product by using functions that use the following commands:
 - Save Licensed Program (SAVLICPGM)
 - Restore License Program (RSTLICPGM)
 - Check Product Option (CHKPRDOPT)
 - Delete License Program (DLTLICPGM)
5. Changing the product object description
This packages the product loads in the product. Once you have packaged the product, you can save it by using the Save License Program (SAVLICPGM) command. You can restore it by using the Restore License Program (RSTLICPGM) command.

Note: It is useful to put the product packaging commands in a source file for use as a CL program or batch job. The create commands contain a large amount of information that is more easily managed in a source file member than interactively running the commands.

OS/400 release considerations

Before creating your product, determine which release of the Operating System/400 program you plan to use to support your product. You can save a product to a previous release using the TGTRLS parameter on the Save Licensed Program (SAVLICPGM) command.

For more information about packaging, see “Release considerations” on page 96 and “Getting your system ready to create PTFs” on page 141. You cannot package products with directories and install them on releases earlier than V4R2M0.

Product packaging restrictions

Limitations of product packaging and items not provided by product packaging on the System Manager licensed program include:

- Limitations
 - If your product is multilingual, you can only use the IBM NLVs.
 - Your product cannot share a library with an IBM product or a product from another development organization. This includes secondary language libraries.
 - Every product must have a library for descriptive information. This restriction includes those products that reside in stream files.
 - A product option cannot have more than one root folder.

- If your product has folders or documents, you must follow a specific folder structure. For more information about folders, see “Packaging products that have folders and documents” on page 56
- On a given system, only one product can have a given product identifier and release level.
- A product load may contain either folders or directories. It cannot contain both folders and directories.
- Items not provided
 - IBM service support (*IBMSRV) for non-IBM products. (Only IBM PTFs can be obtained from *IBMSRV.)
 - A library control system.
 - Tracking of folder documents and directory objects.

Step 1: Creating a product definition

A product definition contains general product information and defines the overall product structure. A product definition is an OS/400 object of type *PRDDFN. Each product has one product definition for each release of the product.

Note: Create the product definition in the main (principal) library for your product by using the Create Product Definition (CRTPRDDFN) command.

Some of the information the product definition contains includes:

- Product Identifier

Identifies a product. Product identifiers created using System Manager commands must be exactly seven characters long and in the format nlxxxx:

- The value of n can be 0 through 9.
- The value of l can be uppercase letters A through Z.
- The value of xxxxx can be a combination of uppercase letters A through Z and numbers 0 through 9.

Note: The product identifiers for IBM products do not follow these naming conventions.

Either you or your iSeries server can create a product identifier. If you specify the value *SER, the system creates a product identifier using the system serial number as part of the identifier. You must attempt to provide a unique product ID (PRDID) for your application. It is recommended that you use *SER for this reason.

If you do specify PRDID (*SER), you must specify the remaining two characters using the product identifier suffix (PRDIDSFX) parameter. The completion message, SMU1322, contains the product identifier created by the system.

System-created product identification assigns the same first five characters to all products created on the same system. This reduces the possibility of product identifier conflicts occurring with other development organizations. Also, when product identifiers are assigned a five-character generic identifier, all of your products are displayed together on all displays.

- Product Description

Allows you to include a textual title for your product in the form of a message. You must supply one or more product descriptions in the **product message file** that you must create with the Create Message File (CRTMSGF) command. When the iSeries server lists product identifiers, the product descriptions may also be

included to explain the product identifiers. The display for the Work with Supported Products (WRKSPTPRD) command provides an example of this.

- Release

Allows you to identify multiple versions of the same product. The release level (RLS) parameter is in the format VnRnMx, where Vn is the version, Rn is the release level, and Mx is the modification level. The value *n* can be 0 through 9, and the value *x* can be 0 through 9 or A through Z.

The release specified for your product can differ from the release of the OS/400 program you use to develop your product or your target to run your product on. It is the release of your product.

Consider your long-term release strategy when choosing the release. One suggestion for ordering your release is to make the first release of a product V1R1M0. The next release could be V1R2M0 or V1R1M1.

Depending on what you specify when creating your product, one or more versions of the product may exist on the system at one time.

- Registration Identifier

Identifies a product as belonging to a specific development organization. Products with different registration identifiers cannot be installed into the same libraries.

Use the registration identifier consistently throughout all your products.

- Product Options

Products have parts called options that can be installed separately. The base product is identified as the **base option** (*BASE). A base option is required for each product. Other options are identified by the values 1 through 99. Options provide a way for your customers to install only the parts of the product that they need. A product is made up of one or more options.

The base option is the only option that is required when the product is installed. Additional options do not need to be installed for the product to function.

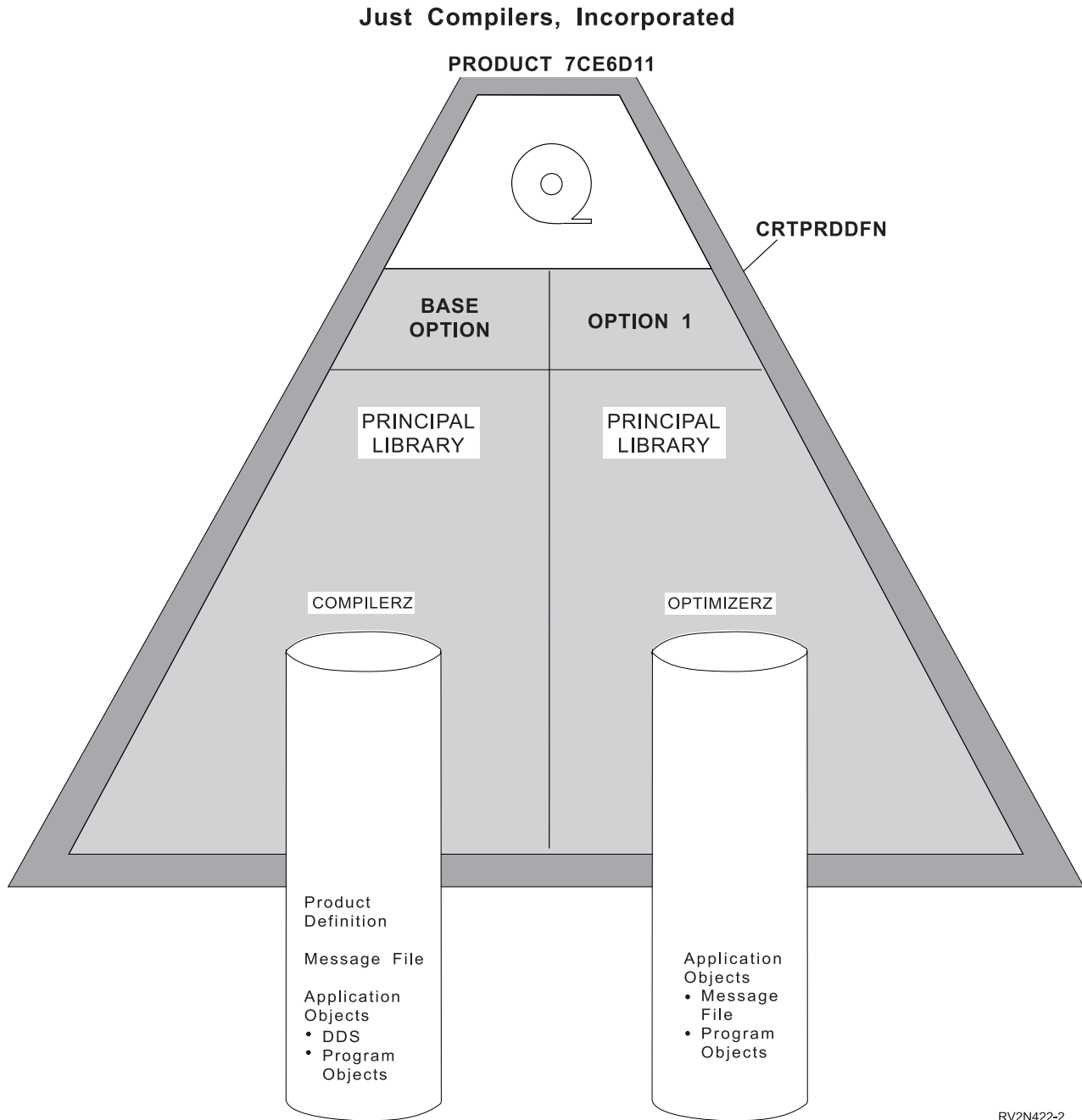
See “Defining features” on page 96 for information about how options are used in creating a license for your product.

When first packaging your product, determine how many options your product has. Also, when planning for a follow-on release, consider whether the new function should be an option and become part of an existing product, or whether it should become a new product. Some factors to consider are:

- An option can have at most 11 libraries.
- An option can have only one root folder. Other folders for the option must be subfolders.
- Options provide a way to make a part of your product separately chargeable.
- Options provide a way for your customers to install only the parts of the product that they need. This saves storage space.
- Allow dynamic naming in the product option if you plan to package directories with your product. See “Establishing a development environment for directories” on page 45.

Product example: Creating a product definition

The following figure shows the structure of a product definition.



RV2N422-2

Figure 9. Creating a Product Example, Step 1

The following example describes how a typical product might appear:

A company, Just Compilers, Incorporated, wants to package an application as a product. The application consists of a compiler and the compiler optimizer. The compiler uses one library, COMPILERZ. The optimizer uses one library, OPTIMIZERZ. Because the compiler is functional with or without the optimizer code, the organization plans to charge an additional fee for the optimizer part of the application.

Some of the objects in the library COMPILERZ include:

- A message file containing the messages for the compiler, including one product description for the compiler and one for the optimizer.
- Data description specification (DDS) for the format of the compiler listing.
- Program objects for the compiler.

Some of the objects in the library OPTIMIZERZ include:

- A message file containing messages for the optimizer.
- Program objects for the optimizer.

Just Compilers, Inc., has decided to package their product as follows:

- The product consists of a base option and option 1. The base option contains the objects needed for the compiler.
- Because the compiler functions without the optimizer, the optimizer can be sold for an additional fee and can exist as Option 1.

The following command creates the product definition for this product:

```

/*****
/* Define the product to the system.          */
*****/

CRTPRDDFN  PRDDFN(COMPILERZ/PRDDFN) +
           PRDID(7CE6D11) +
           RLS(V1R1M0) +
           RGSID(*PHONE 0005075554011) +
           MSGF(CMPMSG) +
           PRDOPT((*BASE MSG0001) +
                 (1  MSG0002)) +
           CPYRGTFST(*CURRENT)

```

After this command is run, the product is defined to your system. Now that the product is defined, messages MSG0001 and MSG0002 are used by commands, such as the Work with Supported Products (WRKSPTPRD) command and the Display Software Resources (DSPSFWRSC) command.

Now additional details about each of the options are needed. Defining additional details is explained in the following section.

Step 2: Creating a product load

Use the Create Product Load (CRTPRDL0D) command to create a product load that ties a group of objects to a product option. The product option may consist of one or more product loads. In this step you define some of the information that the product load contains. This includes the following:

- Libraries for the product option
- Load types
- Directories or folders that belong to the option
- Registration identifier
- Exit programs

Follow the steps below to create your product load.

1. At the command line, type the Create Product Load (CRTPRDL0D) command.
2. In the product load (PRDL0D) parameter field, specify the name for the product load that you are creating.

Notes:

- a. To create the product load object in the library, specify it on the DVLLIB parameter.
- b. The *PRDLOD contains the list of
 - Libraries for the option
 - Folders that belong to the option
 - Directories that belong to the option

Load Types.

There are two types of loads, code and language loads. A **code load** consists of all product objects, such as programs (type *PGM), that are not translated to another language. A code load can contain all the objects for a product or product option. This only occurs if there are no plans to translate any of the objects (such as displays, menus, and messages). A product has one code load for each option.

A **language load** contains product objects to translate into another language if the product is available in more than one language (for example, objects for displays, menus, and messages). For additional information on national language versions, see “National language support” on page 41.

In the load identification parameter field, enter the load ID. This identifies the code load or the language load. Valid code load identifiers are 5001-9999. For additional information on NLVs, see Chapter 2, “Product packaging concepts”.

For information about how you can use code loads when licensing a product, see “Defining features” on page 96.

Product Libraries.

If the objects for a load exist in more than one library, then you must choose one library to be the main library. This library is called the **principal library**. Every product load has a principal library. The principal library contains the product load object. Any other library for that load is called an **additional library**.

For two or more product options to exist in the same library, make sure that the following conditions exist:

- The registration identifier for the product options is the same
- The primary libraries for the options are the same.

For information on additional libraries, see “Packaging products that have more than one library” on page 55.

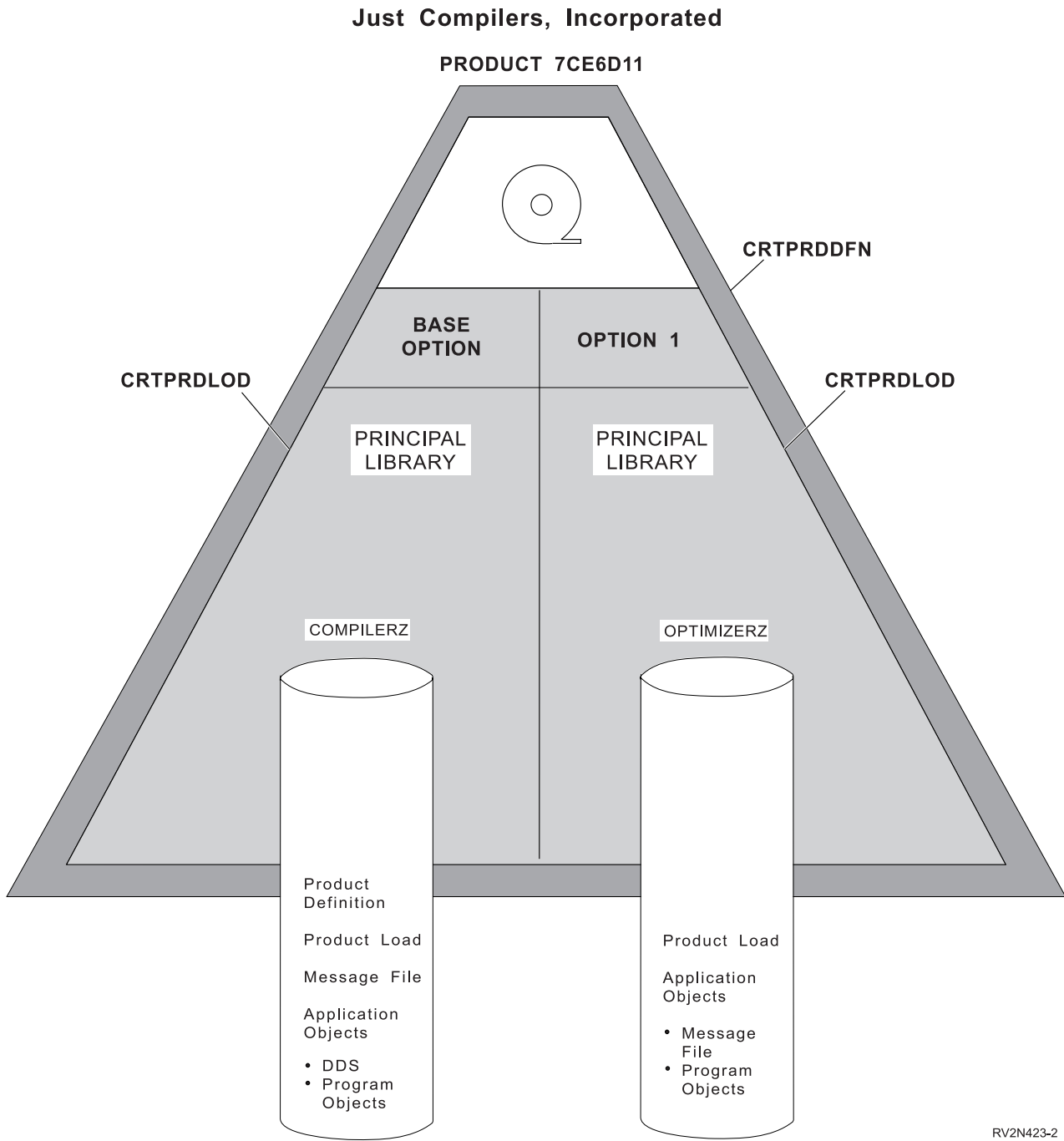
Registration Identifier.

A product load and product definition must have the same registration identifier. As long as you create the product definition first, the *PRDDFN special value provides an easy way to accomplish this. However, if you create the product load object before the product definition, ensure that you use the same registration identifier for both.

Use Create Product Load (CRTPRDLOD) command to create a product load. You can delete the product load by using the Delete Product Load (DLTPRDLOD) command. These are the only two commands you use for product loads.

Product example: Creating a product load

As in the product example, "Product example: Creating a product definition" on page 50, the Create Product Definition (CRTPRDDFN) command creates the product definition for this product. "Step 2: Creating a product load" on page 52 explains how to create the product loads. You can view the results of "Step 1: Creating a product definition" on page 49 and "Step 2: Creating a product load" on page 52 in Figure 10. Figure 10 shows the structure of the product after you created the product definition and product loads.



RV2N423-2

Figure 10. Product Example, Step 2

The following series of commands create the product definition and the product loads for this product:

```

/*****/
/* Define the product to the system.          */
/*****/

CRTPRDDFN PRDDFN(COMPILERZ/PRDDFN) +
          PRDID(7CE6D11) +
          RLS(VIRIM0) +
          RGSID(*PHONE 0005075554011) +
          MSGF(CMPMSG) +
          PRDOPT((*BASE MSG0001) +
                (1  MSG0002)) +
          CPYRGTFST(*CURRENT)

/*****/
/* Provide details about the base option.      */
/*****/

          CRTPRDL0D PRDL0D(PRDL0D) +
          PRDID(7CE6D11) +
          RLS(VIRIM0) +
          OPTION(*BASE) +
          LODTYPE(*CODE) +
          LODID(5001) +
          RGSID(*PRDDFN) +
          DVLLIB(COMPILERZ)

/*****/
/* Provide details about Option 1.            */
/*****/

          CRTPRDL0D PRDL0D(PRDL0D) +
          PRDID(7CE6D11) +
          RLS(VIRIM0) +
          OPTION(1) +
          LODTYPE(*CODE) +
          LODID(5001) +
          RGSID(*PRDDFN) +
          DVLLIB(OPTIMIZERZ)

```

After you run these commands, the system can recognize the product, and the details about each option. Now you must identify the objects that belong to each product load.

Note: For this example product, after you create message file CMPMSG in library COMPILERZ, you must add messages MSG0001 and MSG0002 to the message file.

Packaging products that have more than one library

If you plan to install your product into more than one library, one or more additional libraries can be specified on the Create Product Load (CRTPRDL0D) command. You can specify up to 10 additional libraries for each option. The additional libraries will **not** contain product definitions, product loads, or the product message file.

The controlling parameter is the ADLLIB parameter on the Create Product Load (CRTPRDL0D) command. The following is an example of the changes required to the sample product if option 1 requires an additional library:

```

/*****/
/* Provide details about Option 1.            */
/*****/

```

```

CRTPRDL0D  PRDL0D(PRDL0D) +
            PRDID(7CE6D11) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODTYPE(*CODE) +
            LODID(5001) +
            RGSID(*PRDDFN) +
            DVLLIB(OPTIMIZERZ) +
            ADLLIB((OPTZ2))

```

Packaging products that have folders and documents

If your product contains OS/400 document objects, then those objects must reside in a folder. You specify the root folder and subfolders when you create the product load. Include folders that you want to be a part of the product in the folder list parameter (FLRL) on the Create Product Load (CRTPRDL0D) command. For more information on development folders and primary folders. See Chapter 7, "Servicing your products using System Manager" for information on creating a PTF for documents.

Following are the rules for using folders:

- To perform the packaging function on a product load with folders, make sure that the user performing the packaging is enrolled in the distribution directory.
- Each product option has only one root folder.
- A folder cannot belong to more than one product option.
- The root folder must be part of the code load.
- You must specify folders so that a previous-level folder precedes its subfolder on the list.

For example, you want to include OPTZ and OPTZ/FORMS folders in the folder list. Because OPTZ is a previous-level folder with a subfolder of OPTZ/FORMS, the folders must appear in the list as OPTZ followed by OPTZ/FORMS. It is incorrect to specify OPTZ/FORMS followed by OPTZ.

- If a product has language loads, specify the name of the first folder for a language load must be a root-level folder and a subfolder. For example, OPTN/F0RM2924.

This folder must be a subfolder of the root that is specified for the code load of the product option.

For example, OPTN/F0RM2924 is valid if OPTN is listed for the code load.

- You can specify up to 100 folders for each product load.
- Documents in folders and objects in directories are not specified when a product load is created.

When a product load is packaged, documents in folders and objects in directories are not included in the list of objects created by the packaging function. No list of documents is created or maintained. When the system saves a product by using the Save Licensed Program (SAVLICPGM) command, the documents that are saved are determined as follows: for each folder that is listed when the product load object was created, all documents in that folder are saved.

- When you package a product load, the folders for that load must exist. The user who packages the product load must have *ALL authority to each folder.
- Create all the documents that you want included with your products before you package your product.

You can specify the folder names in the FLRL parameter field on the Create Product Load (CRTPRDLOD) command. The system determines the documents that are saved each time the Save Licensed Program (SAVLICPGM) command runs.

PC code

Products that contain PC code must follow the above rules for folders on the iSeries server. In addition, you create a document named QPTFIDX in each root folder for your product. During PTF processing, PC Support Update (PCSUPDT) function uses the contents of this document. This document triggers the processing performed by PCSUPDT. Therefore, do not use this document to store any information.

Example of packaging a product by using folders

Assume that a release of product 8N01234 exists in the primary libraries and primary folders. Remember that option 1 contains folders. Recall that the primary library names are COMPILERN and OPTIMIZERN, and the primary root folder name is OPTN. The following is an example of creating a second release of product 8N01234 in a set of development libraries and folders. The system creates and packages both the English and Spanish NLVs.

The development libraries for the base option are COMPNDVL1 and COMPNDVL2. The example uses Library COMPNDVL2 to develop the Spanish language load at the same time that the English language load is being developed in library COMPNDVL1. The example uses a separate library for Spanish because the objects for the Spanish language load have the same object names as the objects for the English language load.

For option 1, the development library names are OPTNDVL1 and OPTNDVL2. The example uses OPTNDVL1 to develop the code load and English language load, and uses OPTNDVL2 to develop the Spanish language load. The development folder names are OPTNDVL, OPTNDVL/FORM2924, OPTNDVL/FORM2924/OTHERS, OPTNDVL/FORM2931, and OPTNDVL/FORM2931/OTHERS.

Note that the primary library and primary folder names are the same as the primary names for release of the product.

```

/*****/
/* Define release V4R2M0 of the product to the system. */
/* Languages are English (2924) and Spanish (2931). */
/*****/

CRTPRDDFN PRDDFN(COMPNDVL1/PRDDFN) +
          PRDID(8N01234) +
          RLS(V4R2M0) +
          RGSID(*PHONE 0005072534011) +
          MSGF(CMPMSG) +
          PRDOPT((*BASE MSG0001 *ALWDYNNAM (2924 2931)) +
                (1 MSG0002 *ALWDYNNAM *BASEOPT )) +
          ALWMLTRLS(*YES) +
          CPYRGTFST(*CURRENT)

/*****/
/* Provide details about the base option. */
/* Create a code load and the English language load */
/* into library COMPNDVL1, with a primary library of */
/* COMPILERN for the base option. */
/* Create the Spanish language load into COMPNDVL2, */
/* with a primary library of COMPILERN. */
/*****/

```

```

CRTPRDL0D  PRDL0D(PRDL0D) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODTYPE(*CODE) +
            LODID(5001) +
            RGSID(*PRDDFN) +
            DVLLIB(COMPNDVL1) +
            PRILIB(COMPILERN)

CRTPRDL0D  PRDL0D(LNGL0D) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODTYPE(*LNG) +
            LODID(2924) +
            RGSID(*PRDDFN) +
            DVLLIB(*CODE) +
            PRILIB(*CODE) +
            LNGLIB(COMP2924)

CRTPRDL0D  PRDL0D(LNGL0D) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODTYPE(*LNG) +
            LODID(2931) +
            RGSID(*PRDDFN) +
            DVLLIB(COMPNDVL2) +
            PRILIB(*CODE) +
            LNGLIB(COMP2931)

/*****
/* Mark the objects for release V4R2M0 of the          */
/* base option.                                         */
/* Assume all objects have been created in the         */
/* development libraries for the base option.         */
/* Assume all translatable objects start with 'L' or 'C' */
/* and all other objects start with 'P'.              */
/*****

CHGPRDOBJD  OBJ(COMPNDVL1/P*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(5001)

CHGPRDOBJD  OBJ(COMPNDVL1/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(2924)

CHGPRDOBJD  OBJ(COMPNDVL1/C*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(2924)

CHGPRDOBJD  OBJ(COMPNDVL2/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(2931)

```

```

CHGPRDOB JD OBJ(COMPNDVL2/C*) +
              OBJTYPE(*ALL) +
              PRDID(8N01234) +
              RLS(V4R2M0) +
              OPTION(*BASE) +
              LODID(2931)

/*****
/* Provide details about Option 1. */
/* Create a code load and the English language load */
/* into library OPTNDVL1, with a primary library of */
/* OPTIMIZERN. */
/* Create the Spanish language load into OPTNDVL2, */
/* with a primary library of OPTIMIZERN. */
/* Use the same secondary language libraries as the */
/* base option. */
/* Note that OPTN/FORM2924 is a subfolder of root */
/* folder OPTN. These are primary folders. */
/* Note that OPTNDVL/FORM2924 is a subfolder of root */
/* folder OPTNDVL. These are development folders. */
*****/

CRTPRDLOD PRDLOD(PRDLOD) +
           PRDID(8N01234) +
           RLS(V4R2M0) +
           OPTION(1) +
           LODTYPE(*CODE) +
           LODID(5001) +
           RGSID(*PRDDFN) +
           DVLLIB(OPTNDVL1) +
           PRILIB(OPTIMIZERN) +
           FLRL((OPTNDVL OPTN))

CRTPRDLOD PRDLOD(LNGLODOPT) +
           PRDID(8N01234) +
           RLS(V4R2M0) +
           OPTION(1) +
           LODTYPE(*LNG) +
           LODID(2924) +
           RGSID(*PRDDFN) +
           DVLLIB(*CODE) +
           PRILIB(*CODE) +
           LNLIB(COMP2924) +
           FLRL(('OPTNDVL/FORM2924' +
                'OPTN/FORM2924') +
                ('OPTNDVL/FORM2924/OTHERS' +
                 'OPTN/FORM2924/OTHERS'))

CRTPRDLOD PRDLOD(LNGLODOPT) +
           PRDID(8N01234) +
           RLS(V4R2M0) +
           OPTION(1) +
           LODTYPE(*LNG) +
           LODID(2931) +
           RGSID(*PRDDFN) +
           DVLLIB(OPTNDVL2) +
           PRILIB(*CODE) +
           LNLIB(COMP2931) +
           FLRL(('OPTNDVL/FORM2931' +
                'OPTN/FORM2931') +
                ('OPTNDVL/FORM2931/OTHERS' +
                 'OPTN/FORM2931/OTHERS'))

/*****
/* Mark the objects for release V4R2M0 of option 1. */
/* Assume all objects have been created in the */
/* development libraries for option 1. */
*****/

```

```

/* Assume all translatable objects start with 'L'. */
/*****

CHGPRDOBJD OBJ(OPTNDVL1/P*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODID(5001)

CHGPRDOBJD OBJ(OPTNDVL1/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODID(2924)

CHGPRDOBJD OBJ(OPTNDVL2/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODID(2931)

/*****
/* Create folders OPTNDVL, OPTNDVL/FORM2924, */
/* OPTNDVL/FORM2924/OTHERS, OPTNDVL/FORM2931, */
/* and OPTNDVL/FORM2931/OTHERS. */
/* Also create documents within the folders. */
/*****
.
.
.

/*****
/* Add the product descriptions prior to packaging. */
/* Messages are needed in both English and Spanish for */
/* both the compiler and the optimizer. */
/*****

ADDMSGD MSGID(MSG0001) +
        MSGF(COMPNDVL1/CMPMSG) +
        MSG('Compiler N')

ADDMSGD MSGID(MSG0002) +
        MSGF(COMPNDVL1/CMPMSG) +
        MSG('Compiler N Optimizer')

ADDMSGD MSGID(MSG0001) +
        MSGF(COMPNDVL2/CMPMSG) +
        MSG('Compilador N')

ADDMSGD MSGID(MSG0002) +
        MSGF(COMPNDVL2/CMPMSG) +
        MSG('Compilador N Optimizador')

/*****
/* Package the code load, the English load, and the */
/* Spanish load for each option. LODID(*ALL) is the */
/* default for the PKGPRDOPT command. */
/*****

PKGPRDOPT PRDID(8N01234) +
          RLS(V4R2M0) +
          OPTION(*BASE) +
          ALWAPICHG(*SAME)

PKGPRDOPT PRDID(8N01234) +

```

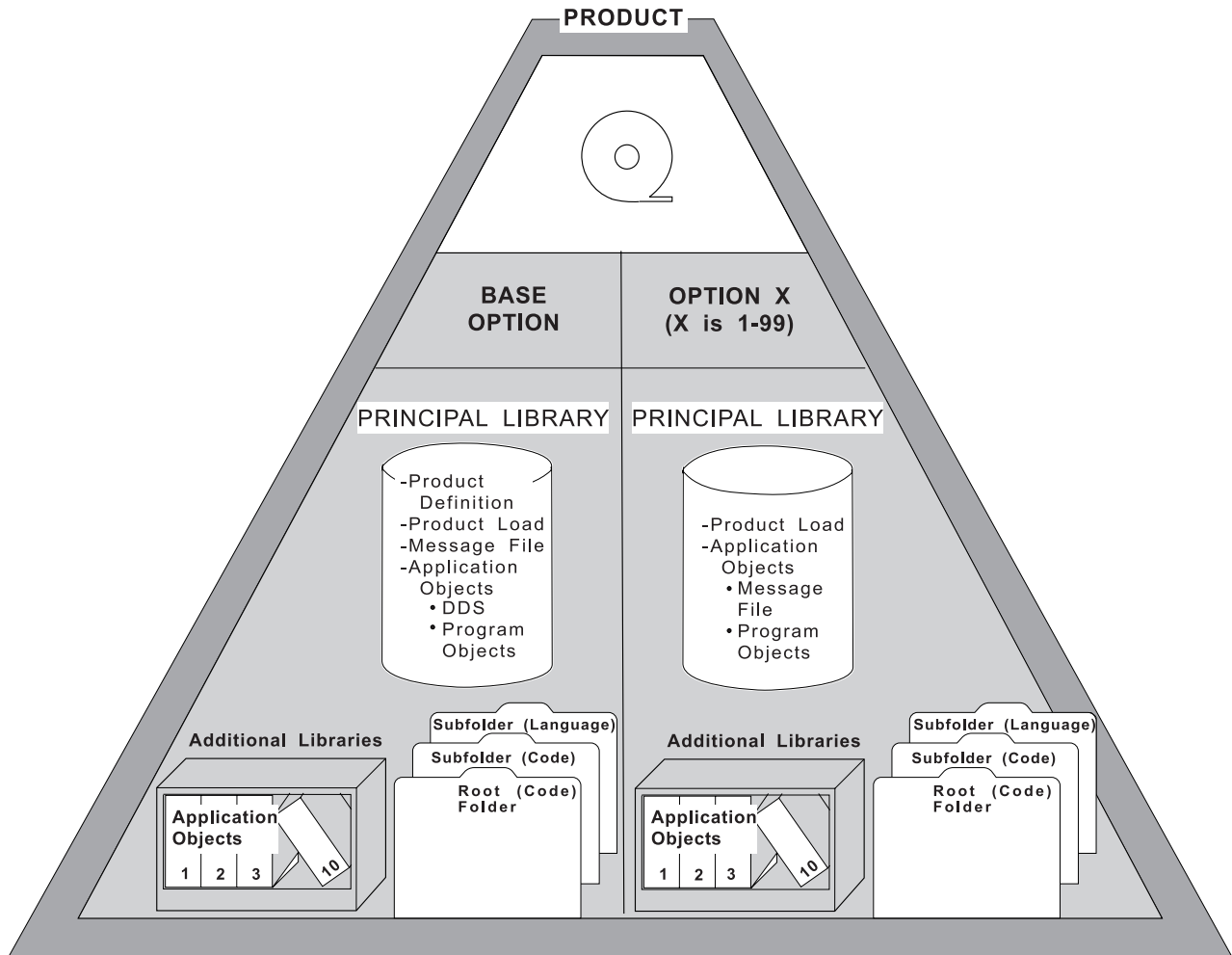
```

RLS(V4R2M0) +
OPTION(1) +
ALWAPICHG(*SAME)

```

It is important that the language product load for the base option (LNGLOD) and the language product load for option 1 (LNGLODOPT) have different object names. This allows you to install both options in Spanish and English at the same time. The product load objects for the two options can both exist in the secondary language library.

Figure 11 shows objects that are associated with packaging a product with folders.

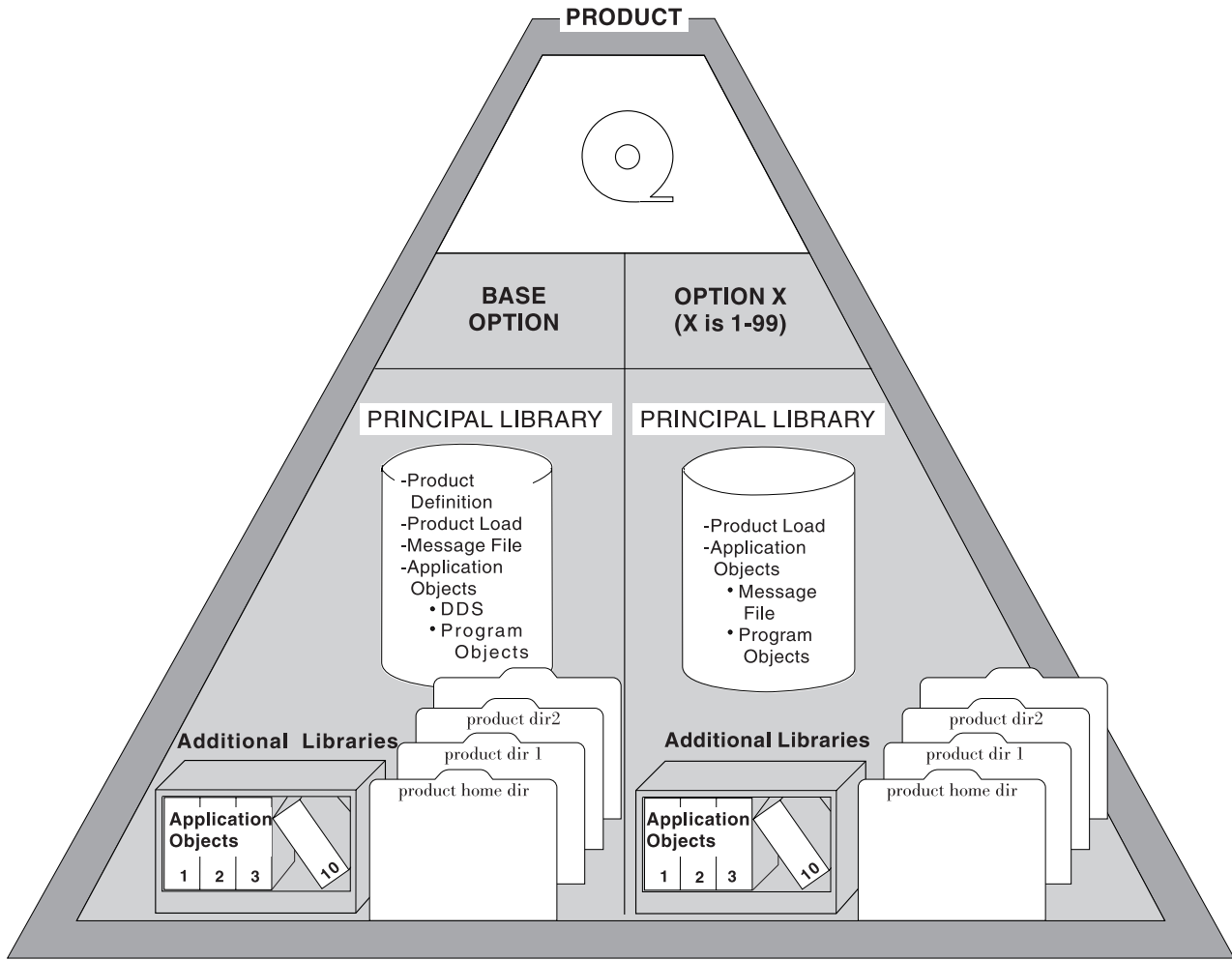


RV2N416-7

Figure 11. Packaging a Product with Folders

Packaging products that have directories

If you have packaged your product option by using folders, you cannot create your product load to support directories for the current version. Remember that a product option may have folders or directories but not both of them.



RV4N355-2

Figure 12. Packaging a Product with Directories

Dynamic naming conventions for directories

A directory path is one or more directories and sub-directories that reads from left to right. For example, consider a path name for a file that is named /World/Location/State/files shipped with an LP. The path name consists of two types of directories, product home directories and product directories. The product home directories are represented on the left side of the path name. One or more product loads can share a product home directory. In the example the path /World/Location represents product home directories. The Integrated File System files that are shipped with the LP are the files that reside in the product directory /State.

Product directories are located on the right side of the product home directory name. Product directories do not share product loads across multiple products. /State is an example of a product directory. The product directory path name contains both the product home directories and product directories. The full path cannot be shared across product loads.

You must decide if your product will have product home directories. One advantage of having product home directories is that product home directories

names can be reassigned by using the Restore Licensed Program (RSTLICPGM) command. As a result, your iSeries server can automatically manage them for you.

Assigning names to directories in the product load

To specify the home and product directories that are assigned to your product load, type the Create Product Load (CRTPRDLOD) command. In the directory list (DIRL) parameter field, enter the name of your directories. You can specify up to a combined sum of 300 product home directories and product directories. Be sure that your directory names contain the following:

- Precede your path name with a forward slash (/).
- Name your path names using upper case (A-Z) and lower case (a-b) alphabets and numeric values (0-9).
- You can use the following symbols in the directory names:
 - Comma (,)
 - Period (.)
 - Colon (:)
 - Semi Colon (;)
 - Ampersand (&)
 - Plus sign (+)
 - Forward slash (/)
 - Percent sign (%)
 - Equal sign (=)
 - Greater than sign (>)
 - Less than sign (<)
 - Right Parenthesis)
 - Left Parenthesis (
 - Minus sign (-)
 - Underscore (_)

Following are some restrictions for directories:

- You cannot specify /QSYS.LIB and /QDLS directories for the DIRL parameter.
- Do not end the path name with a forward slash (/).
- Do not use a blank in the directory path name.
- Do not use the following character combinations of ../ and ./ for any directory path names.

Example of packaging a product by using directories

Assume that a release of product 8N01234 exists in the primary libraries and primary folders. Recall that the primary library names are COMPILERN and OPTIMIZERN, and the primary root folder name is OPTN. The following is an example of creating a second release of product 8N01234 in a set of development libraries and directories. The system creates and packages both the English and Spanish NLVs. This allows you to manage your product by directories on the current system release.

The development libraries for the base option are COMPNDVL1 and COMPNDVL2. The example uses Library COMPNDVL2 to develop the Spanish language load at the same time that the English language load is being developed in library COMPNDVL1.

For option 1, the development library names are OPTNDVL1 and OPTNDVL2. The example uses OPTNDVL1 to develop the code load and English language load, and uses OPTNDVL2 to develop the Spanish language load. Directories reside in the root file system. The directories have a product home directory name of OPTN. The product home directory is now OPTN/CODE, OPTN/CODE/OTHERS, OPTN/FORM2924, OPTN/FORM2924/OTHERS, OPTN/FORM2931, and OPTN/FORM2931/OTHERS.

Consider creating product option 1 to allow dynamic naming. See “Establishing a development environment for directories” on page 45.

```

/*****
/* Define release V4R2M0 of the product to the system. */
/* Languages are English (2924) and Spanish (2931). */
/*****

CRTPRDDFN PRDDFN(COMPNDVL1/PRDDFN) +
          PRDID(8N01234) +
          RLS(V4R2M0) +
          RGSID(*PHONE 0005072534011) +
          MSGF(CMPMSG) +
          PRDOPT((*BASE MSG0001 *ALWDYNNAM (2924 2931)) +
                (1 MSG0002 *ALWDYNNAM *BASEOPT )) +
          ALWMLTRLS(*YES) +
          CPYRGTFST(*CURRENT)

/*****
/* Provide details about the base option. */
/* Create a code load and the English language load */
/* into library COMPNDVL1, with a primary library of */
/* COMPILERN for the base option. */
/* Create the Spanish language load into COMPNDVL2, */
/* with a primary library of COMPILERN. */
/*****

CRTPRDLOD PRDLOD(PRDLOD) +
          PRDID(8N01234) +
          RLS(V4R2M0) +
          OPTION(*BASE) +
          LODTYPE(*CODE) +
          LODID(5001) +
          RGSID(*PRDDFN) +
          DVLLIB(COMPNDVL1) +
          PRILIB(COMPILERN)

CRTPRDLOD PRDLOD(LNGLOD) +
          PRDID(8N01234) +
          RLS(V4R2M0) +
          OPTION(*BASE) +
          LODTYPE(*LNG) +
          LODID(2924) +
          RGSID(*PRDDFN) +
          DVLLIB(*CODE) +
          PRILIB(*CODE) +
          LNGLIB(COMPNDVL1)

CRTPRDLOD PRDLOD(LNGLOD) +
          PRDID(8N01234) +
          RLS(V4R2M0) +
          OPTION(*BASE) +
          LODTYPE(*LNG) +
          LODID(2931) +
          RGSID(*PRDDFN) +
          DVLLIB(COMPNDVL2) +
          PRILIB(*CODE) +
          LNGLIB(COMPNDVL2)

/*****

```

```

/* Mark the objects for release V4R2M0 of the          */
/* base option.                                         */
/* Assume all objects have been created in the         */
/* development libraries for the base option.         */
/* Assume all translatable objects start with 'L' or 'C' */
/* and all other objects start with 'P'.             */
/*****

```

```

CHGPRDOBJD OBJ(COMPNDVL1/P*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(5001)

```

```

CHGPRDOBJD OBJ(COMPNDVL1/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(2924)

```

```

CHGPRDOBJD OBJ(COMPNDVL1/C*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(2924)

```

```

CHGPRDOBJD OBJ(COMPNDVL2/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(2931)

```

```

CHGPRDOBJD OBJ(COMPNDVL2/C*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(*BASE) +
            LODID(2931)

```

```

/*****
/* Provide details about Option 1.                    */
/* Create a code load and the English language load   */
/* into library OPTNDVL1, with a primary library of   */
/* OPTIMIZERN.                                         */
/* Create the Spanish language load into OPTNDVL2,   */
/* with a primary library of OPTIMIZERN.             */
/* Use the same secondary language libraries as the   */
/* base option.                                        */
/* Note that OPTN is a product home directory for the */
/* product directories, CODE, CODE/OTHERS, FORM 2924, */
/* FORM 2924/OTHERS, FORM 2931, FORM 2931/OTHERS    */
/*****

```

```

CRTPRDLOD PRDLOD(PRDLOD) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODTYPE(*CODE) +
            LODID(5001) +
            RGSID(*PRDDFN) +
            DVLLIB(OPTNDVL1) +
            PRILIB(OPTIMIZERN) +
            DIRL(('OPTN' ('/CODE' '/CODE/OTHERS')))

```

```

CRTPRDL0D  PRDL0D(LNGL0DOPT) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODTYPE(*LNG) +
            LODID(2924) +
            RGSID(*PRDDFN) +
            DVLLIB(*CODE) +
            PRILIB(*CODE) +
            LNGLIB(COMP2924) +
            DIRL((' /OPTN' ('FORM2924' FORM2924' +
                        'FORM2924/OTHERS'))))

CRTPRDL0D  PRDL0D(LNGL0DOPT) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODTYPE(*LNG) +
            LODID(2931) +
            RGSID(*PRDDFN) +
            DVLLIB(OPTNDVL2) +
            PRILIB(*CODE) +
            LNGLIB(COMP2931) +
            DIRL((' /OPTN' ' /FORM2931' +
                        ' /FORM2931/OTHERS'))))

/*****
/* Mark the objects for release V4R2M0 of option 1.      */
/* Assume all objects have been created in the           */
/* development libraries for option 1.                   */
/* Assume all translatable objects start with 'L'.       */
*****/

CHGPRDOBJD  OBJ(OPTNDVL1/P*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODID(5001)

CHGPRDOBJD  OBJ(OPTNDVL1/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODID(2924)

CHGPRDOBJD  OBJ(OPTNDVL2/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V4R2M0) +
            OPTION(1) +
            LODID(2931)

/*****
/* Create directories OPTN/CODE, OPTN/CODE,             */
/* OPTN/CODE/OTHERS, OPTN/F0RM2931,                     */
/* OPTN/F0RM2929, OPTN/2929/OTHERS                       */
/* OPTN/F0RM2931/OTHERS, OPTN/F0RM2931.                 */
/* OPTN/F0RM                                           */
/* Also create files within the directories              */
/* Note that OPTN is a home directory and cannot        */
/* contain files for the product.                       */
*****/

```

```

/*****
/* Add the product descriptions prior to packaging. */
/* Messages are needed in both English and Spanish for */
/* both the compiler and the optimizer. */
/*****

ADDMSGD MSGID(MSG0001) +
        MSGF(COMPNDVL1/CMPMSG) +
        MSG('Compiler N')

ADDMSGD MSGID(MSG0002) +
        MSGF(COMPNDVL1/CMPMSG) +
        MSG('Compiler N Optimizer')

ADDMSGD MSGID(MSG0001) +
        MSGF(COMPNDVL2/CMPMSG) +
        MSG('Compilador N')

ADDMSGD MSGID(MSG0002) +
        MSGF(COMPNDVL2/CMPMSG) +
        MSG('Compilador N Optimizador')

/*****
/* Package the code load, the English load, and the */
/* Spanish load for each option. LODID(*ALL) is the */
/* default for the PKGPRDOPT command. */
/*****

PKGPRDOPT PRDID(8N01234) +
          RLS(V4R2M0) +
          OPTION(*BASE) +
          ALWAPICHG(*SAME)

PKGPRDOPT PRDID(8N01234) +
          RLS(V4R2M0) +
          OPTION(1) +
          ALWAPICHG(*SAME)

```

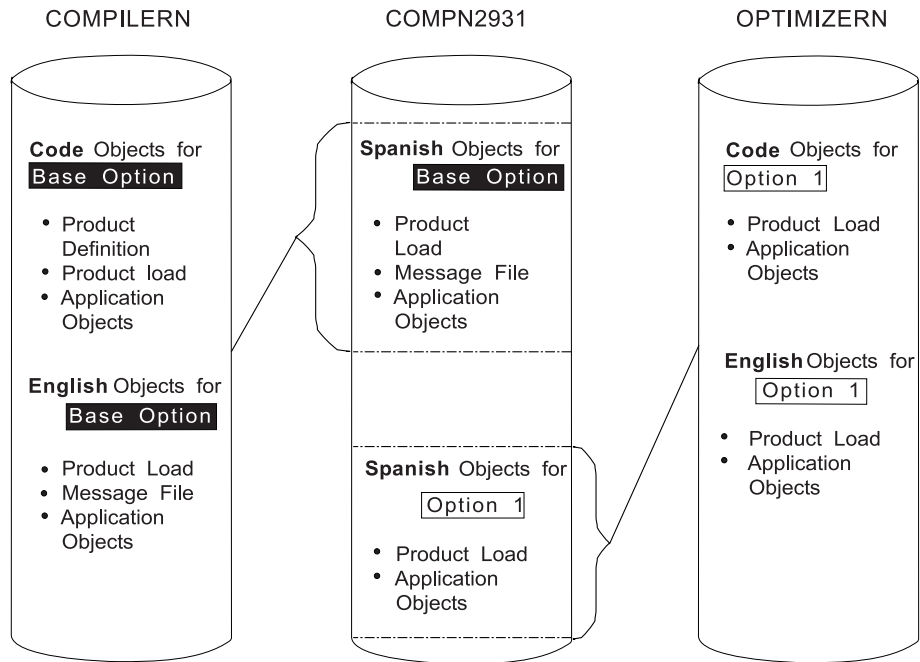
It is important that the language product load for the base option (LNGLD) and the language product load for option 1 (LNGLDOPT) have different object names. This allows you to install both options in Spanish and English at the same time. The product load objects for the two options can both exist in the secondary language library.

Figure 12 on page 62 shows objects that are associated with packaging a product with directories.

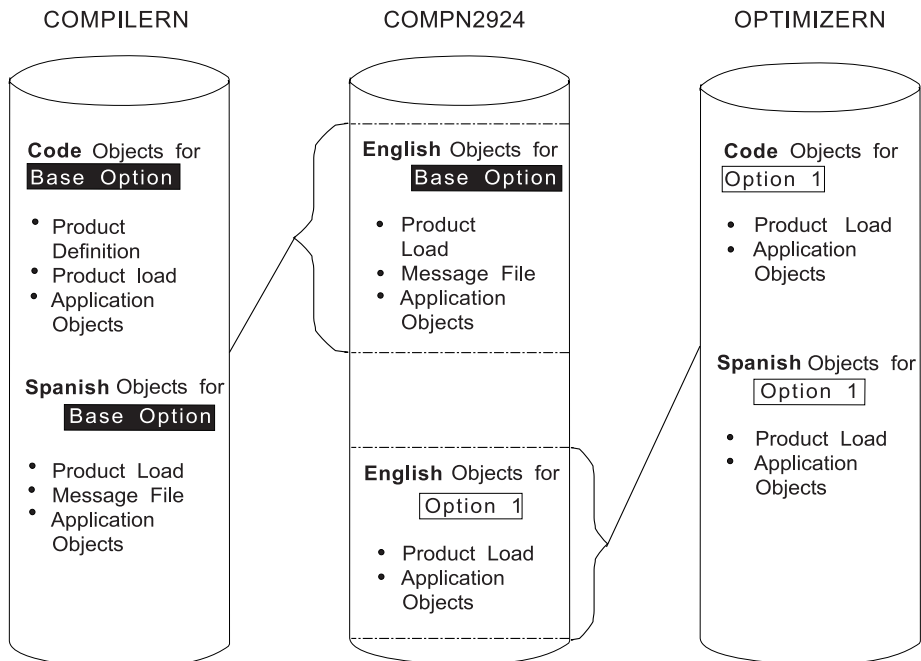
Understanding national language loads for your product

The language load contains all the product objects that you will be translating. Figure 13 on page 68 shows where product objects exist after the product is installed on two systems. One system has English as the primary language, and the other system has Spanish as the primary language. The language identifier for English is 2924 and for Spanish is 2931.

ENGLISH SYSTEM



SPANISH SYSTEM



RV2N427-6

Figure 13. Product Objects for a Product Installed on Two Systems

NLVs for all options must be listed for the base option. If an option other than the base option is available for a given NLV, the NLV must be identified as a language for the base option.

For example, if the base option is available in English and Spanish, and Option 1 is available in French, then French must also be listed as an NLV for the base option. You would specify *BASEOPT for Option 1.

Specify NLV information on the fourth element of the PRDOPT parameter. This parameter is on the Create Product Definition (CRTPRDDFN) command. To specify the NLVs for an option other than the base option, one of the following must be specified:

- *BASEOPT
The NLVs specified for the base can be used.
- *NONE
This option is not multilingual.

If the base product has language features, the product message file must be marked to be an object in the language load rather than the code load. This is done using the Change Product Object Description (CHGPRDOBJD) command. This allows a message file to be created for each language, and for message files to be installed in correct libraries like all other language objects.

If the option is available in only one language, language loads are not needed. The language objects may be included with the code portion of the option. However, if the product could become available in more than one language before your next release of the product, a list of language loads might be needed. The code load must be packaged or installed before the associated language loads can be packaged.

Secondary language library

The language part of an option is installed on a system into one of the following:

- Product option principal and additional libraries
- Secondary language library

If the NLV of an option being installed is the same as the primary language of the system, the language is installed into the product option principal and additional libraries. If the language (NLV) is different than the primary language of the system, it is installed into a secondary language library.

A secondary language library name is specified using the LNGLIB parameter on the Create Product Load (CRTPRDL0D) command. Each NLV for an option should have its own secondary language library. Options of a product may share secondary language libraries. Depending on the registration identifier specified, products can share secondary language libraries.

Objects that are part of a secondary language load cannot be installed into a library that is part of a packaged code load. For example, if the primary language of the system is Spanish, the English objects for the product (secondary language load objects) cannot be installed into a library that contains code load objects of a packaged product.

Object names

Each object for a language load should have the same object name for each NLV. Similarly, the product load object in each NLV for a given product option must have the same object name. This ensures that language objects correctly reinstall if the primary language of a system is changed and a product is reinstalled. In this

case, the language objects for the new primary language will replace the language objects for the old primary language. This is especially important for the name of the product load objects.

Secondary language libraries and object names

Several products or several options of the same product can share a secondary language library. If two libraries share a secondary language library, you must ensure that no two language load objects use the same name for different options. If two language objects use the same name, one replaces the other if both are installed into the same secondary language library.

Note: Objects associated with a language load can be created in more than one product library. For a language installed as a secondary language, objects from the principal and additional libraries are installed into only one library, the secondary language library. You must specify the name of the secondary language library when a language load object is created.

Folder Names

When creating more than one language load for an option, the only folder name that should be changed is the subfolder of the root folder for the option.

The following is an example of a product that will be translated into Spanish, but will first be available only in English:

```

/*****/
/* Define the product to the system.          */
/* Languages are English (2924) and Spanish (2931). */
/*****/

CRTPRDDFN PRDDFN(COMPILELN/PRDDFN) +
          PRDID(8N01234) +
          RLS(V1R1M0) +
          RGSID(*PHONE 0005072534011) +
          MSGF(CMPMSG) +
          PRDOPT((*BASE MSG0001 *NODYNNAM (2924 2931)) +
                (1 MSG0002 *NODYNNAM *BASEOPT )) +
          CPYRGTFST(*CURRENT) +
          ALWMLTRLS(*YES)

/*****/
/* Provide details about the base option.      */
/* Create a code load and the English language load. */
/*****/

CRTPRDL0D PRDL0D(PRDL0D) +
          PRDID(8N01234) +
          RLS(V1R1M0) +
          OPTION(*BASE) +
          LODTYPE(*CODE) +
          LODID(5001) +
          RGSID(*PRDDFN) +
          DVLLIB(COMPILELN)

CRTPRDL0D PRDL0D(LNGL0D) +
          PRDID(8N01234) +
          RLS(V1R1M0) +
          OPTION(*BASE) +
          LODTYPE(*LNG) +
          LODID(2924) +
          RGSID(*PRDDFN) +
          DVLLIB(*CODE) +
          LNGLIB(COMP2924)

```



```

/*****/
/* Mark the objects for the base option. */
/* Assume all translatable objects start with 'L' or 'C' */
/* and all other objects start with 'P'. */
/*****/

CHGPRDOBJD OBJ(COMPILELN/P*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V1R1M0) +
            OPTION(*BASE) +
            LODID(5001)

CHGPRDOBJD OBJ(COMPILELN/L*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V1R1M0) +
            OPTION(*BASE) +
            LODID(2924)

CHGPRDOBJD OBJ(COMPILELN/C*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V1R1M0) +
            OPTION(*BASE) +
            LODID(2924)

/*****/
/* Provide details about Option 1. */
/* Create a code load and the English language load. */
/* Use the same secondary language library as the */
/* base option. */
/* Note that OPTN/FORM2924 is a subfolder of root */
/* folder OPTN. */
/*****/

CRTPRDLOD PRDLOD(PRDLOD) +
            PRDID(8N01234) +
            RLS(V1R1M0) +
            OPTION(1) +
            LODTYPE(*CODE) +
            LODID(5001) +
            RGSID(*PRDDFN) +
            DVLLIB(OPTIMIZERN) +
            FLRL((OPTN))

CRTPRDLOD PRDLOD(LNGLODOPT) +
            PRDID(8N01234) +
            RLS(V1R1M0) +
            OPTION(1) +
            LODTYPE(*LNG) +
            LODID(2924) +
            RGSID(*PRDDFN) +
            DVLLIB(*CODE) +
            ENGLIB(COMP2924) +
            FLRL(('OPTN/FORM2924') +
                ('OPTN/FORM2924/OTHERS'))

/*****/
/* Mark the objects for option 1. */
/* Assume all translatable objects start with 'L'. */
/*****/

CHGPRDOBJD OBJ(OPTIMIZERN/P*) +
            OBJTYPE(*ALL) +
            PRDID(8N01234) +
            RLS(V1R1M0) +
            OPTION(1) +

```

```

                                LODID(5001)

CHGPRDOBJD  OBJ(OPTIMIZERN/L*) +
                                OBJTYPE(*ALL) +
                                PRDID(8N01234) +
                                RLS(V1R1M0) +
                                OPTION(1) +
                                LODID(2924)

/*****
/* Add the product descriptions prior to packaging.      */
*****/

ADDMSGD     MSGID(MSG0001) +
            MSGF(COMPILERN/CMPMSG) +
            MSG('Compiler N')

ADDMSGD     MSGID(MSG0002) +
            MSGF(COMPILERN/CMPMSG) +
            MSG('Compiler N Optimizer')

/*****
/* Package each option.                                  */
*****/

PKGPRDOPT  PRDID(8N01234) +
            RLS(V1R1M0) +
            OPTION(*BASE) +
            ALWAPICHG(*SAME)

PKGPRDOPT  PRDID(8N01234) +
            RLS(V1R1M0) +
            OPTION(1) +
            ALWAPICHG(*SAME)

```

The folder list for the Spanish language load is:

```

FLRL(('OPTN/FORM2931') +
      ('OPTN/FORM2931/OTHERS'))

```

The name of the first folder for the Spanish load is the root folder with a different subfolder than was used by the English load. This is to allow both the English and the Spanish folders to exist on the system together.

It is important that the language product load for the base option (LNGLOD) and the language product load for option 1 (LNGLODOPT) have different object names so that when both options are installed on an iSeries server with a primary language of Spanish, both objects can exist in the secondary language library, COMPN2924.

Step 3: Packaging products that have exit programs

You can call an exit program either before or after a specific operation. Exit programs generally perform setup or cleanup functions.

If you are planning to package your product with directories, consider writing your exit program in a programming language that allows parameter lists which are larger than 75K. CL restricts parameter lists to 10K. In many cases, CL's 10K limitation is not sufficient when working with directories.

One or more exit programs can be defined for a product. The exit programs assist with the install, save, delete, and check product functions.

Exit programs must be listed in the product load object for the code or in the product load object for a language and must be associated with a particular product library. The exit programs must exist in the library where they are used.

This section discusses the differences between preoperation and postoperation exit programs as well as how to complete the following:

- Product exit program interfaces
- Exit program functions for saving a product
- Exit program functions for restoring a product
- Exit program functions for deleting a product
- Exit program functions for checking a product
- Exit program rules and restrictions

Preoperation exit program

During certain product operations, such as installing the product option, some tasks must be performed before the operation takes place. For example, if you are installing a product option on a system that has an earlier version of the product already installed, you may need to convert a file that contains customer information into a format that the new release of the product requires. You can use a preoperation exit program for this conversion. The operations that follow allow preoperation processing:

- Saving a product using the Save Licensed Program (SAVLICPGM) command
- Restoring a product using the Restore Licensed Program (RSTLICPGM) command
- Checking a product using the Check Product Option (CHKPRDOPT) command
- Deleting a product using the Delete License Program (DLTLICPGM) command

A preoperation exit program corresponds to the library where it exists. A list of preoperation programs can exist for each library in a product load. Before these operations are performed, the system calls its first postoperation exit program in the list for that product library. This program then calls other preoperation exit programs on the list. You can list up to ten preoperation exit programs.

Note: The list includes all the programs that are called during the preoperation processing. The system handles these programs specially to ensure that they exist when the system needs them. If you use the Restore Licensed Program (RSTLICPGM) command to install the products, remember that several things happen as the system restores each library. The system restores preoperation exit programs into the QTEMP library first. Then they are called before the objects are restored into that product library.

Postoperation exit program

During certain product operations, such as installing the product option, you may need to perform some tasks after the operation takes place. For example, after installing the product, a user profile must be created. Use a postoperation exit program to perform such an operation. The following operations allow postoperation processing:

- Saving a product using the Save Licensed Program (SAVLICPGM) command
- Restoring a product using the Restore Licensed Program (RSTLICPGM) command
- Checking a product using the Check Product Option (CHKPRDOPT) command

- Deleting a product using the Delete License Program (DLTLICPGM) command

A postoperation exit program corresponds to the library where it exists. If a product load has more than one library, a postoperation program can exist for each library. After these operations are performed, the postoperation exit program is called. This program may then call any other product program objects. You can name one postoperation exit program. Unlike the preoperation programs, the object that the load needs already exists on the system.

Product exit program interface

For information about the interface between the exit program and the Restore Licensed Program (RSTLICPGM), Save Licensed Program (SAVLICPGM), Delete Licensed Program (DLTLICPGM), and Check Product Option (CHKPRDOPT) commands, see the API topic under the Programming category in the Information Center.

Error messages: The following messages are the **only** error messages that should be sent from the product management exit programs.

- CPF3D95 - Exit program processing failed.
If an error occurs in an exit program, the exit program must send the message CPF3D95 as an exception. This message should be sent when your exit program is unable to do a function such as duplicate an object, rename an object, or function check, for example. The caller of the exit program monitors for this message and leaves it in the job log. The message indicates to the caller that an error was in the exit program and that processing for the product did not complete.
- CPF3D98 - Exit program processing found error in product.
This message should be sent only for the *CHKCODE, *CHKLNG, *SAVCODE, or *SAVLNG functions to indicate that an error was detected in a product. Before sending this message, you should send a message indicating what you found wrong with the product, for example, objects were missing (see message below) or not configured correctly.
- CPD3DC8 - &5; &6; in &7; not found for product &1; option &2; release &4;
This message should be sent by an exit program when it determines that a missing object is a critical object, which means the product is not complete without it. (This is only used by the *SAVCODE, *SAVLNG, *CHKCODE, and *CHKLNG functions, because they are the only ones checking for critical missing objects.)
CPF3D98 must be sent as a final exception when CPD3DC8 is sent.
- CPD3DE7 - Folder &6; not found for product &1; option &2; release &4;
This message should be sent by an exit program when it determines that a missing folder is a critical folder, which means the product is not complete without it. (This is only used by the *SAVCODE, *SAVLNG, *CHKCODE, and *CHKLNG functions, because they are the only ones checking for critical missing folders.)
CPF3D98 must be sent as a final exception when CPD3DE7 is sent.
- CPD3D70 - Directory &6; not found for product &1; option &2; release &4;
This message should be sent by an exit program when it determines that a missing directory is a critical directory, which means the product is not complete without it. (This is only used by the *SAVCODE, *SAVLNG, *CHKCODE, and *CHKLNG functions, because they are the only ones checking for critical missing directories.)

Exit program functions for saving a product

The following summarizes the responsibilities of product management exit programs while saving products:

- Verify missing objects, folders, or directories. A list of missing objects, folders, or directories is passed to your exit program. You need to determine if it is acceptable for these objects to be missing. Send message CPD3DC8, CPD3DE7, or CPD3D70 for each missing object, folder, or directory that the user needs to be made aware of.

Note: If you do not have an exit program, missing objects, folders, or directories are verified for you.

If you have a postoperation exit program, your exit program must verify missing objects, folders, or directories because the Save Licensed Program (SAVLICPGM) command does not verify them for you.

- Verify the existence of your documents if your product contains documents. The list of missing objects that is passed to your exit program does not include documents or stream files. You must provide your exit program with the names of any documents or stream files that your exit program should check.
- Perform any special setup procedures necessary to ensure that the objects saved from the product libraries can be used to successfully install the product.
- Communicate errors to calling program. See “Error messages” on page 74 for more information on messages.

Exit program functions for restoring a product

The following are some of the functions that might be performed by a product exit program during product restoration and installation:

- Delete objects from prior releases of the product that are obsolete in the current release.
- Perform any special setup procedures to ensure that all objects in the product library can be successfully restored.

For example, assume a database file already exists on a system. This database file was created at a different time than the database file being restored. During the restore function, the database file on the system will be renamed, and two database files will exist on the system. It may be necessary for the exit program to delete old database files to ensure extra database files are not created during a restore function.

- Create any objects required by the product that cannot be saved using the Save Object (SAVOBJ) command.

The Save Licensed Program (SAVLICPGM) command uses the SAVOBJ command. If an object cannot be saved using the SAVOBJ command, the object cannot be saved using the SAVLICPGM command, either.

- Grant private authorizations to user profiles for any installed objects.

If an object does not already exist on the system, public authority to the restored object must match the public authority that existed when the object was saved. If the object already exists on the system, public authority is unchanged by restoring a new version of the object. User-defined authorities are never saved or restored with the object.

If any objects for the product option require user defined authority for specific user profiles, these user-defined authorities must be granted by the exit program. For example, any command that is not authorized to the public must be privately authorized to specific user profiles by an exit program.

- Perform any special processing necessary to install language objects in a secondary language library.
As an example, your product contains some database files that must only exist in library PRDLIB. The same distribution package is used for both primary and secondary language orders, so the exit program for your product language objects must delete the version of these special files that is automatically restored from the distribution tape into the secondary language library by the Restore Licensed Program (RSTLICPGM) command.
- Communicate errors to calling program. See “Error messages” on page 74 for more information on messages.

Exit program functions for deleting a product

The following summarizes the responsibilities of product management exit programs during product deletion:

- If your program creates objects in other libraries, delete these objects.
- Communicate errors to calling program. See “Error messages” on page 74 for more information on messages.

Exit program functions for checking a product

The following summarizes the responsibilities of product management exit programs while checking products:

- Verify missing objects, folders, or directories. A list of missing objects, folders, and directories is passed to your exit program. You need to determine if it is acceptable for these objects to be missing. Send message CPD3DC8, CPD3DE7, or CPD3D70 for each missing object, folder, or directory that the user needs to be made aware of.
- Verify the existence of your directories if your product contains documents. Verify the existence of your stream files if your product contains documents. The list of missing objects that is passed to your exit program does not include documents or stream files. You must provide your exit program with the names of any documents or stream files that your exit program should check.
- Verify that any setup processing performed by your installation exit program was successful.

Note: If you do not have an exit program, missing objects, folders, and directories are verified for you.

If you have a postoperation exit program, your exit program must verify missing objects and folders. This is because the Check Product Option (CHKPRDOPT) command does not verify them for you.

- Communicate errors to calling program. See “Error messages” on page 74 for more information on messages.

Exit program rules and restrictions

This section summarizes important rules and restrictions for product management exit programs:

- Product exit programs must be prepared to run in an environment where required products are not installed.
 - An exit program for code load objects cannot assume the existence of language load objects for the product.
 - An exit program for one product library cannot assume the existence of libraries that are packaged in a different product. This is because products can be installed in any order.

- An exit program cannot assume the existence of libraries that are packaged in any other option for the product.
- The order in which the product libraries were entered when the Create Product Load (CRTPRDLOD) command was used is significant. A library that occurs in this list after the library for which the exit program is running might not exist yet.

This is because the options for a product may be installed in any order. The libraries within a product option are saved or restored in the order they appear in the list of libraries contained in the product load object and are deleted in inverse order.

- An exit program cannot assume that any product library (including the library that contains it) is in the job library list. The library that contains the exit program is passed as a parameter to the exit program.
- Product exit programs must not use any function that only works in an interactive job.

The Save Licensed Program (SAVLICPGM), Restore Licensed Program (RSTLICPGM), Delete Licensed Program (DLTLICPGM), and Check Product Option (CHKPRDOPT) commands operate in batch as well as interactively, so there might be no interactive display device for the job when the exit program is called.

- Product exit programs must not save or restore objects directly to or from media. Saving or restoring a complete product option requires a series of save or restore operations. The Operating System/400 program must have control over these operations (device selection, tape file positioning, and so on) in order for the operations to work correctly.
- An exit program is required if you want to delete folders or directories when you are installing a second release.
- Product exit programs must avoid designs that require the user to provide any information other than that provided on the CL command during product save, restore, delete, or check operations.

Product exit programs are intended to install, save, delete, or check a product; they are not to be used for tailoring the product or system.

Note: Although it may be possible for a preoperation exit program to display or request data, the associate display files would not be present at this time.

- Exit programs must monitor for their own messages. If any errors occur during the running of the exit program and the operation should not continue, see "Error messages" on page 74 for appropriate messages to send.

Exit programs should not put messages in the job log that the customer does not need to see. For example, if the exit program looks for an object and does not care if the object does not exist, the exit program should remove the "object not found" message from the job log.

- Exit programs run with the authority of the user profile that is calling the Restore Licensed Program (RSTLICPGM), Save Licensed Program (SAVLICPGM), Delete Licensed Program (DLTLICPGM), or Check Product Option (CHKPRDOPT) command.
- Having product exit programs places the responsibility of missing-object processing on the exit programs. This processing includes sending messages for missing objects, folders, and directories. If your product does not have exit programs, the Save Licensed Program (SAVLICPGM) command and the Check Product Option (CHKPRDOPT) command send the messages for you.

Note: If your product has exit programs and they do not process the missing-object list, missing-folder list, or missing-directory list, problems with missing objects for your product will not be noticed. In this case, the product might appear to be correctly installed when it is not installed.

Step 4: Creating message files and messages

Before you mark your objects for packaging, you must create your message files. To create messages, use the Create Message File (CRTMSGF) command. When the iSeries server lists product identifiers, the product descriptions may also be included to explain the product identifiers. The display for the Work with Supported Products (WRKSPTPRD) command provides an example of this.

Step 5: Marking the objects for packaging

Object information is associated with each object on the system. This information is referred to as the **object description**. The following information in the object description is used for product packaging:

- Product identifier
- Product release level
- Product option
- Product load identifier

Marking objects for packaging involves changing these fields in the description of an object in order to supply this information.

The Change Product Object Description (CHGPRDOBJD) command is used to change this information in an object description.

Each object, with the exception of libraries, folders, and documents, must have the correct object description information if it is to be packaged with a product. This includes the product definition (*PRDDFN), the product load (*PRDLOD) objects, the product message file, and the product exit programs.

Note: Every object in every library associated with a product option must have the CHGPRDOBJD command run. This includes objects in additional libraries.

Product example: Marking objects for packaging

Figure 14 on page 79 shows the objects used in step 3 of the product example.

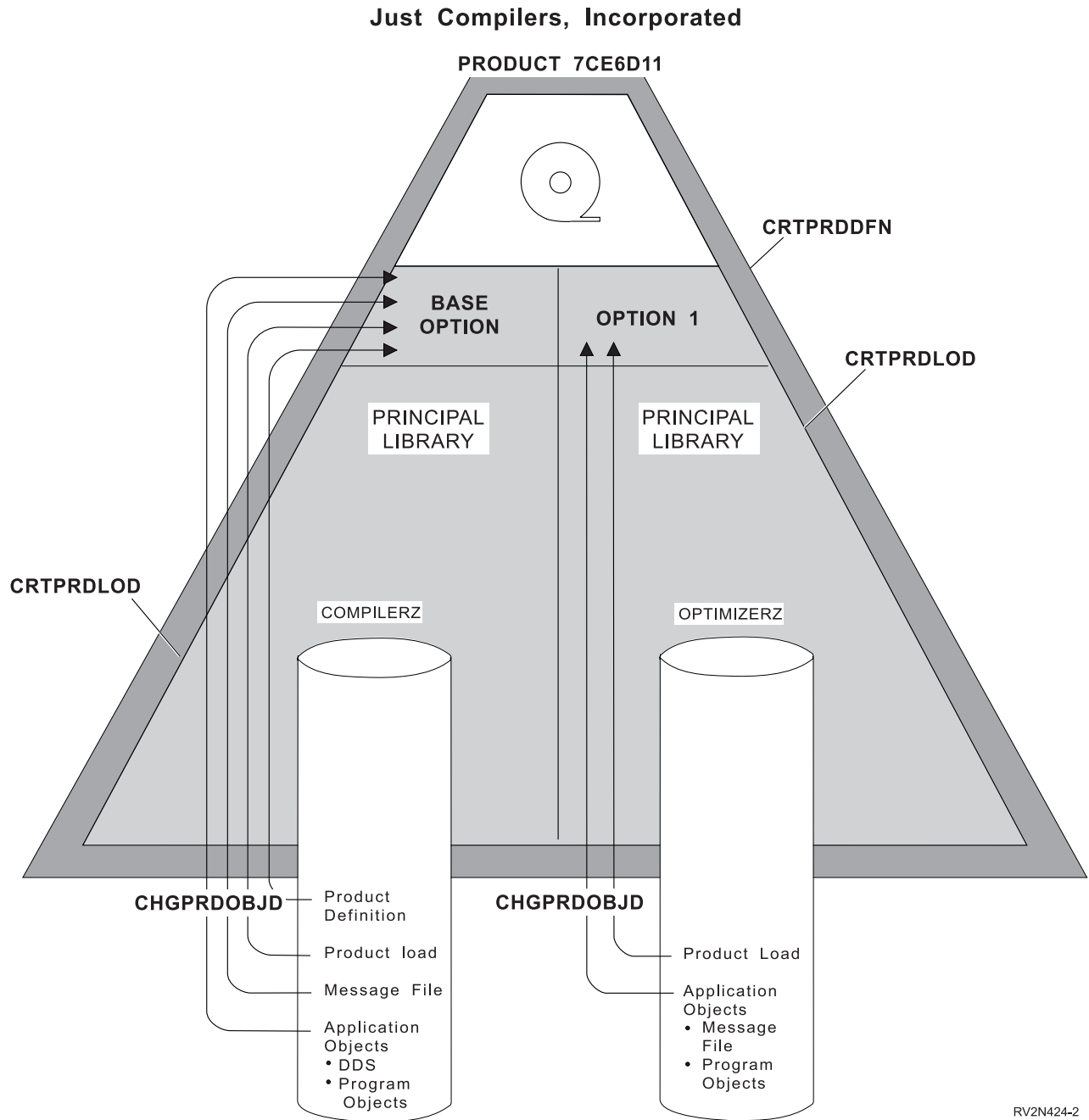


Figure 14. Product Example, Step 3

The following example shows how to change product object descriptions for products packaged for the iSeries server:

```

/*****
/* Define the product to the system. */
/*****

CRTPRDDL PRDDL(COMPILERZ/PRDDL) +
PRDL(7CE6D11) +
RLS(V1R1M0) +
RGSID(*PHONE 0005075554011) +
MSGF(CMPMSG) +

```

```

        PRDOPT((*BASE MSG0001) +
              (1   MSG0002)) +
        CPYRGTFST(*CURRENT)

/*****
/* Provide details about the base option.          */
*****/

CRTPRDL0D  PRDL0D(PRDL0D) +
           PRDID(7CE6D11) +
           RLS(V1R1M0) +
           OPTION(*BASE) +
           LODTYPE(*CODE) +
           LODID(5001) +
           RGSID(*PRDDFN) +
           DVLLIB(COMPILERZ)

/*****
/* Mark the objects for the base option.          */
*****/

        CHGPRDOBJD OBJ(COMPILERZ/*ALL) +
        OBJTYPE(*ALL) +
        PRDID(7CE6D11) +
        RLS(V1R1M0) +
        OPTION(*BASE) +
        LODID(5001)

/*****
/* Provide details about Option 1.              */
*****/

CRTPRDL0D  PRDL0D(PRDL0D) +
           PRDID(7CE6D11) +
           RLS(V1R1M0) +
           OPTION(1) +
           LODTYPE(*CODE) +
           LODID(5001) +
           RGSID(*PRDDFN) +
           DVLLIB(OPTIMIZERZ)

/*****
/* Mark the objects for Option 1.              */
*****/

CHGPRDOBJD  OBJ(OPTIMIZERZ/*ALL) +
            OBJTYPE(*ALL) +
            PRDID(7CE6D11) +
            RLS(V1R1M0) +
            OPTION(1) +
            LODID(5001)

```

After these commands are run, the product is defined to your system, the details about each of the options is known to the system, and the objects belonging to each product load have been identified. Now the objects need to be packaged which is explained in the following section.

Step 6: Packaging your product option

Packaging is the step that associates the product objects with a product load. The objects that have been marked for packaging are associated with a product load when that load is packaged. Each product load must be packaged before it can be distributed. The command that is used to package a product load is the Package Product Option (PKGPRDOPT) command. The only object types that the PKGPRDOPT command packages are those object types that are valid on the Save Object List API. Other object types are ignored.

A packaging request operates either on all product loads associated with an option, or on a single product load. If a product has more than one option and the entire product is to be packaged, the command must be issued once for each product option.

For the packaging step to be successful, the following conditions must be met:

- The product definition object must exist in the main library (principal library) for the code load for the base option.
- The product load object must exist in the main library (principal library) of that load.
- The product option being packaged must have been specified when the product definition was created.
- All product libraries defined for the product load must exist.
- When the base option is packaged, the product message file must exist in the correct library.
- When the base option is packaged, the product descriptions (one message per option) must exist in the product message file.
- All exit programs must exist in the correct libraries.
- No library to be packaged can be secured by an authorization list.
- The public authority to each library to be packaged must not be user defined; it must be a special value, such as *CHANGE.

Note: The base product (*BASE) need not be packaged with its options; you can package an option as long as the product definition has been created.

Installing the product can cause the product libraries to be created. Before the Package Product Option (PKGPRDOPT) command is used, the library type (*PROD or *TEST), public authority, and create authority for the libraries must be set to the values needed when the product is installed.

Note: The Package Product Option (PKGPRDOPT) command stores the library type (*PROD or *TEST), public authority, and create authority of the libraries for later use by the Restore Licensed Program (RSTLICPGM) command. If your product requires specific values for these items, ensure they are correct before using the Package Product Option (PKGPRDOPT) command.

When the Package Product Option (PKGPRDOPT) command is run, messages are issued if any of the above conditions are not met. When all of the above conditions are met, the objects in the product libraries for the option or load being packaged are checked to determine to what product, release, option, and load they belong. This is done by looking at the object description for each object in these product libraries. As objects are found that belong to the option or load being packaged, the name and type of the object is added to a list created by the packaging function. If a load has additional libraries, a list is created for each library. The objects in these lists make up the product load.

The following restrictions apply:

- The list includes only product load objects that were created to package the option. Product load objects are not included in the list even if the object descriptions indicate that other product load objects should be included.
- The list includes only the product definition object that was created for the release of the product being packaged, even if the object descriptions indicate that other product definition objects should be included.

- A product cannot include a product availability object (type *PRDAVL).

If additional objects are added to the product libraries after the Package Product Option (PKGPRDOPT) command has been run, these additional objects are not part of the product load unless you run the command again and specify REPACKAGE(*YES). The REPACKAGE parameter prevents successfully packaged product loads from being accidentally packaged again. If a load has already been packaged and needs to be packaged again, specify REPACKAGE(*YES).

Note: Installed products can be repackaged by specifying REPACKAGE(*YES). Take care when overriding the default value. The product will no longer appear to the system as installed after being repackaged.

If you do not want to allow changes to the object descriptions after the objects have been packaged, set an indicator when the product load is packaged. The allow API change (ALWAPICHG) parameter of the Package Product Option (PKGPRDOPT) command controls this indicator. This indicator prevents the Change Product Object Description (CHGPRDOBJD) command or the QLICOBJD application program interface (API) from changing the object descriptions of the objects packaged.

After the Package Product Option (PKGPRDOPT) command runs, save the product by using the Save Licensed Program (SAVLICPGM) command. Using the Restore Licensed Program (RSTLICPGM) command, you can restore the product to a system with OS/400 program at the same or later version, release, and modification level as that of the system you used to package your product.

Note: The Save Licensed Program (SAVLICPGM) command uses the Save Object (SAVOBJ) command. If an object cannot be saved using the SAVOBJ command, the object cannot be saved using the SAVLICPGM command.

Product example: Packaging your product option

Figure 15 on page 83 shows the objects used in step four of the product example.

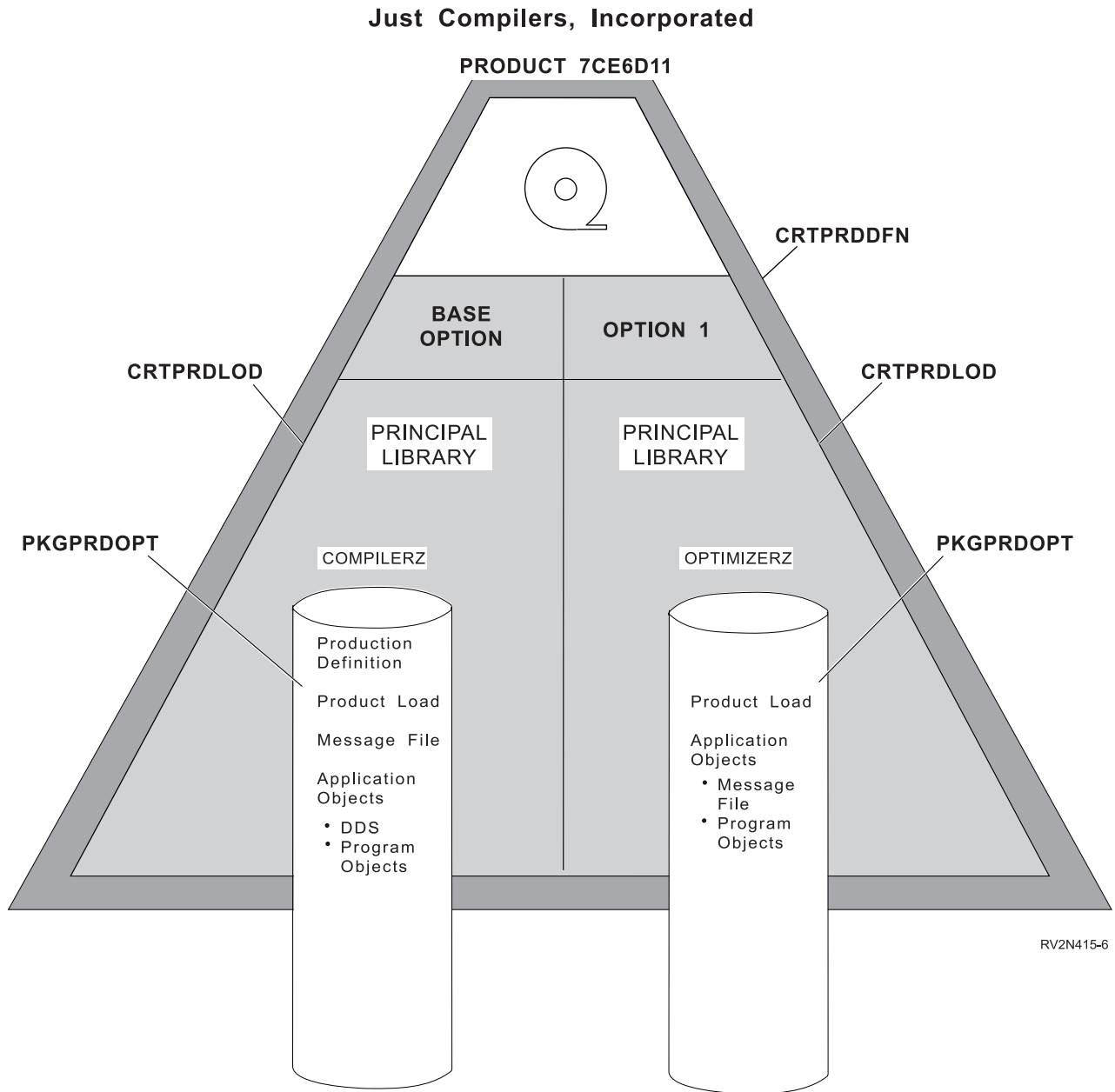


Figure 15. Product Example, Step 4

This part of the example shows how to finish packaging your product.

```

/*****
/* Define the product to the system.          */
*****/

CRTPRDDFN PRDDFN(COMPILERZ/PRDDFN) +
          PRDID(7CE6D11) +
          RLS(V1R1M0) +
          RGSID(*PHONE 0005075554011) +
          MSGF(CMPMSG) +
          PRDOPT((*BASE MSG0001) +
                (1  MSG0002)) +
          CPYRGTFST(*CURRENT)

```

```

/*****
/* Provide details about the base option.          */
*****/

CRTPRDLOD  PRDLOD(PRDLOD) +
           PRDID(7CE6D11) +
           RLS(V1R1M0) +
           OPTION(*BASE) +
           LODTYPE(*CODE) +
           LODID(5001) +
           RGSID(*PRDDFN) +
           DVLLIB(COMPILERZ)

/*****
/* Mark the objects for the base option.          */
*****/

CHGPRDOBJD OBJ(COMPILERZ/*ALL) +
           OBJTYPE(*ALL) +
           PRDID(7CE6D11) +
           RLS(V1R1M0) +
           OPTION(*BASE) +
           LODID(5001)

/*****
/* Provide details about Option 1.                */
*****/

CRTPRDLOD  PRDLOD(PRDLOD) +
           PRDID(7CE6D11) +
           RLS(V1R1M0) +
           OPTION(1) +
           LODTYPE(*CODE) +
           LODID(5001) +
           RGSID(*PRDDFN) +
           DVLLIB(OPTIMIZERZ)

/*****
/* Mark the objects for Option 1.                */
*****/

CHGPRDOBJD OBJ(OPTIMIZERZ/*ALL) +
           OBJTYPE(*ALL) +
           PRDID(7CE6D11) +
           RLS(V1R1M0) +
           OPTION(1) +
           LODID(5001)

/*****
/* Add the product descriptions prior to packaging.  */
*****/

ADDMSGD    MSGID(MSG0001) +
           MSGF(COMPILERZ/CMPMSG) +
           MSG('Compiler Z')

ADDMSGD    MSGID(MSG0002) +
           MSGF(COMPILERZ/CMPMSG) +
           MSG('Compiler Z Optimizer')

/*****
/* Package each option.                          */
*****/

PKGPROPT   PRDID(7CE6D11) +
           RLS(V1R1M0) +
           OPTION(*BASE) +
           ALWAPICHG(*SAME)

PKGPROPT   PRDID(7CE6D11) +

```

```
RLS(V1R1M0) +  
OPTION(1) +  
ALWAPICHG(*SAME)
```

The product is now ready to be saved using the Save Licensed Program (SAVLICPGM) command.

Step 7: Saving and restoring your product

It is important that you save and restore your product by using the Save License Program (SAVLICPGM) and Restore License Program (RSTLICPGM) commands. If you do not save and restore your product on your iSeries server, you cannot create or load apply PTFs for it.

Saving Your License Program

Use the Save License Program (SAVLICPGM) command to save the product options that exist on the system. This command also saves all the loads or parts of the loads that are contained in the product option. To save the loads of an object, you must do at least one of the following tasks:

- To save all the code and language loads of an option, specify *ALL on the object type (OBJTYPE) parameter of the Save License Program (SAVLICPGM) command.
- To save the code load of an option, specify *PGM on the object (OBJTYPE) parameter of the Save License Program (SAVLICPGM) command.
- To restore a specific language load of an option, specify *LNG on the object type (OBJTYPE) parameter of the Save License Program (SAVLICPGM) command. Be sure to specify the language ID in the language (LNG) parameter.

Note: The object type (OBJTYPE) parameter field does not refer to OS/400 object types. Instead, it refers to load types. Remember that there are two types of loads, code and language loads. See “Step 2: Creating a product load” on page 52.

Restoring your license program

Use the Restore License Program (RSTLICPGM) command to restore the product options that exist on the system. This command also restores all the loads or subsets of loads that are contained in the product option. Remember that an option of a product cannot be installed before the base option of the product is installed. To restore the loads of an object, you must do at least one of the following tasks:

- To restore both the code and the primary language load of an option, specify *ALL on the type of object to be restored (RSTOBJ) parameter of the Restore License Program (RSTLICPGM) command.
- To save the code load of an option, specify *PGM on the type of object to be restored (RSTOBJ) parameter of the Restore License Program (RSTLICPGM) command.
- To restore a specific language load of an option, specify *LNG type of object to be restored (RSTOBJ) parameter of the Save License Program (RSTLICPGM) command. Be sure to specify the language type in the language (LNG) parameter.

Note: The object type (RSTOBJ) parameter field does not refer to OS/400 object types. Instead, it refers to load types. Remember that there are two types of loads, code and language loads. See “Step 2: Creating a product load” on page 52.

Step 8: Checking your product structure

The CL command, Check Product Option (CHKPRDOPT), should be used to determine what is wrong with the structure of a product when problems occur. The information it reports indicates differences between the packaged product structure and the current structure of the product.

The CHKPRDOPT command checks the following:

- The product load (PRDLOD) object
- The product definition (PRDDFN) object, if the product load being checked is the base code load
- Libraries
- Directories
- Folders
- Objects (excluding documents in folders) belonging to the product load.
- If the product load has a postoperation exit program, the exit program may perform other checks.

Typically, the informational or diagnostic messages sent by the Check Product Option (CHKPRDOPT) command must be used to determine the condition of a product. For example, the Check Product Option (CHKPRDOPT) command does not necessarily issue an escape message when checking a product that has been deleted.

The CHKPRDOPT command verifies that the structure and objects of a product load are correct. The CHKPRDOPT command does not verify that products function correctly. The CHKPRDOPT command sends a message when it detects an error, at which time it will continue processing and report as many errors as possible.

The following are restrictions for the CHKPRDOPT command

- It does not check documents in folders.
- It does not check for stream files in directories.
- It does not check all objects for the operating system.
- It reports renamed objects as missing.
- It reports an object as missing if the product ID, release level, option, and load ID in the object description do not match the product load.
- It does not report an error if a licensed program is at a previous release level from that of the operating system.
- The Check Product Option (CHKPRDOPT) command sends a message for each defined product load that exists. To get the defined message, specify the detail (DETAIL) parameter field with a value of *FULL.
- If an exit program fails with CPF3D95 or CPF3D98 (the messages that are expected from an exit program), CHKPRDOPT will not send messages for the missing object or folder list entries. It assumes that the exit program may have already sent messages for missing objects or folders. In cases where the missing objects are allowed to be missing, it would be inappropriate for CHKPRDOPT to send messages.

Note: If an exit program signals an unexpected exception, CHKPRDOPT reports the error and sends messages for the missing object or folder entries.

Product packaging summary

Table 3 summarizes how each item relates to each command. Refer to this table when using packaging commands. The table displays items that are related to product packaging in the column on the left. The product packaging commands are shown across the top.

Table 3. Product Packaging Summary

	CRTPRDDFN	CRTPRDLOD	CHGPRDOBJD	PKGPRDOPT	SAVLICPGM	RSTLICPGM
Product Message File	Name must be specified Message ID must be specified for each product option		Must exist	Message file must exist Message descriptions should be added		
Exit Programs		Names must be specified Number of preoperation exit programs must be known	Must exist	Must exist	Preoperation programs called Postoperation programs called	Preoperation programs called Postoperation programs called
Product Definition			Must exist	Must exist regardless of which option is being packaged		
Product Load Object			Must exist	Must exist for product, release, option, load being packaged Not necessary that *PRDLODs for other options have been created		
Secondary Language Library		Must be specified				Used if installing a language other than the primary language of the system
Development Library	Principal development library for base code load	Principal must exist		Library attributes: TYPE, AUT, CRTAUT stored		

Table 3. Product Packaging Summary (continued)

	CRTPRDDFN	CRTPRDLOD	CHGPRDOBJD	PKGPRDOPT	SAVLICPGM	RSTLICPGM
Primary Library		Must be specified				Name used if one of the following is true: <ul style="list-style-type: none"> • Dynamic naming not allowed • Not replacing another release and LIB parameter not specified
Development Folder		Must be specified. Additional libraries must be specified.		Must exist If language load, first folder name must be subfolder of code's root folder		
Primary Folder		Must be specified		If language load, first folder name must be subfolder of code's root folder		Name used if one of the following is true: <ul style="list-style-type: none"> • Dynamic naming not allowed • Not replacing another release and FLR parameter not specified
Document				Should be created for size calculation	Must exist to be saved	
Other Functional Objects			Must exist	Must exist		
Directories	Dynamic naming recommended	must be specified				

Table 3. Product Packaging Summary (continued)

	CRTPRDDFN	CRTPRDLOD	CHGPRDOBJD	PKGPRDOPT	SAVLICPGM	RSTLICPGM
Registration ID	Must be specified	Must be specified				Cannot be restored if product with different registration ID exists in libraries or folders
Dynamic Naming	Must be specified for each option					Used to determine whether library and folder names can be overridden
Allow Multiple Releases	Must be specified					Used to prevent additional releases from being installed
NLVs	Must be specified for each option	Identified by LODID for language loads	Identified by LODID for language loads	Identified by LODID for language loads	Identified by LNG for language loads	Identified by LNG for language loads

Commonly asked product packaging questions

What software is enabled for OS/400 PTFs?

Any software that has been packaged by the System Manager.

What objects can be part of a PTF?

Any object that meets the following restrictions can be part of a PTF:

- You must be able to rename the object to a 10 character name. For example, *GSS object types can only be eight characters long. Therefore these object types cannot be part of a PTF.
- You must be able to save and restore the object.
- You must be able to duplicate the object.
- You must be able to move the object.
- You cannot include a product definition (*PRDDFN) as part of a PTF.
- You cannot include a product load (*PRDLOD) as part of a PTF.

What solutions can be provided by a PTF?

A fix to a known product problem and a functional upgrade to an existing product.

How do I handle source code, if I make it available?

Source code, if made available, can often be in a separate option because not all customers want a copy of the source code. Be careful when creating PTFs for source code you ship if your customers customize that code. This presents a special set of problems for application maintenance. When you send a maintenance fix for the function code, the customer must redistribute and change the source for the functional code.

Can I use PTFs to track things like changes to application programs, changes to rate tables, or changes to database formats?

Yes.

Does packaging change my application functionally?

No.

Will I have to recompile my application when I package it?

No.

Do I need multiple iSeries servers to develop my products?

The product must be installed to create PTFs for it. Depending on the product, you may want more than one system.

Do I need to create a product load for every option?

Yes. At least a code load needs to be created for every option.

Do I need to create a message file for every option?

No. A message file is required by System Manager only for the base option. A message for each option should exist in that message file.

Can more than one product exist in the same library?

If the registration identifiers for the products are the same, they may be installed into the same library. Otherwise, they cannot.

Can more than one option of a product exist in the same library?

Yes, if the primary library names for the options are the same.

Can more than one product use the same secondary language library?

If the registration identifiers for the products are the same, they may be installed into the same library. Otherwise, they cannot.

Can more than one option of a product use the same secondary language library?

Yes, if the secondary language library names, specified when the product was created, are the same.

Can two products have documents in the same folder or directory?

This cannot be done with packaging commands. This can only be done by exit programs.

Can two options have documents in the same folder or directory?

This is not recommended. This can only be done by exit programs.

Can a product that is packaged with System Manager be distributed with Management Central?

Yes, they both use the same underlying APIs.

Can a product that is packaged with Management Central be distributed with System Manager?

Yes, they both use the same underlying APIs.

Do I need to create a library to package my PC product?

Yes, if you want to distribute your product using an iSeries server you need to create a library, a product definition, at least one code load, and a product message file to contain the product descriptions.

What if one of my product libraries is in use by another packaged product?

If your product allows dynamic naming, new library names can be specified on the Restore Licensed Program (RSTLICPGM) command. If your product does not allow dynamic naming, the only way for your product to be installed is if the registration identifier of your product matches the registration identifier of the product already in the library.

How do I handle objects that cannot be restored over themselves?

The system does not allow objects of certain types, such as journals, journal receivers, folders, and data queues, to restore over themselves. Therefore, if such an object is part of a product, the product requires preoperation exit programs. The preoperation exit programs must delete the objects to be able to restore a product over itself. Make sure your users understand that they should not have other objects or data in your folders or directories because the system deletes the objects or data when the product is installed again.

How do I handle objects that cannot be restored into a library?

The system does not allow objects of certain types, such as journals, journal receivers, folders, and data queues, to restore over themselves. If a product includes one of these objects, put it in an option that does not allow dynamic naming, or create it separately by an exit program.

Can I install multiple products at one time?

You can use the Load and Run (LODRUN) command to provide your customers with a way to install more than one of your products. By using the LODRUN command, you can customize the installation of your products.

What are considerations for object ownership and authorizations?

You need to ensure that all referenced user profiles exist on all target systems. This may require a program to create a new user profile that can be run by the user or the install exit program. Authorizations could also be granted by this program.

|
|
|
|
|

Chapter 4. Packaging products for licensing

User-based pricing allows a customer to pay for a product based on the number of users. A license management function is provided in the OS/400 licensed program to monitor usage. Read this chapter if you want to use the OS/400 license management function.

User-based software pricing has advantages for the following customer needs:

Customer demand

Customers are asking for software pricing alternatives that equate price paid to value received.

Customer value

The value of a particular software package is often associated with the number of users of that package, regardless of the size of the model on which it runs.

Customer flexibility

Customers are able to select the pricing option that best satisfies their needs in a cost-effective way.

The iSeries Access product supports user-based pricing. The APIs associated with this product also allow other PC-based products to follow user-based pricing procedures.

Overview of license management

A license gives the user authorization to use a product. For products that have a user-based charge option, a customer may choose to pay for use of the license based on the number of authorized users (authorized usage limit).

You must define license usage information by using the Add Product License Information (ADDPRDLICI) command so the OS/400 license management function can monitor usage of your product. The ADDPRDLICI command is used in conjunction with the software packaging commands discussed in Chapter 3, "Packaging your products" on page 47 and Chapter 2, "Product packaging concepts" on page 37.

The license usage information is packaged and shipped with your product and includes the usage type, compliance type, license term and default usage limit. Once your product is installed, the license owner sets the authorized usage limit.

A programming interface is provided so that your product can request and release a use of the license. The use of the product is assigned to a license user until it is released. To remain in compliance with the license, the number of users must not exceed the authorized usage limit.

The license management function assists customers in managing license use by:

- Monitoring usage levels.
- Providing notification to the customer when approaching and exceeding authorized usage limits.

- Printing a form for ordering additional usage authorization to mail to the software provider.
- Allowing the customer to set and change authorized usage limits.

Planning

Consider pricing, revenue, and marketing strategy when determining whether your product should be user-based priced. Legal counsel should participate in the planning stage because the use of software is usually regulated by legal agreements. Pricing software as user-based may also mean changes in the way the product is developed and shipped. The following decisions should result from the planning stage:

- Ordering process changes
- License term strategy
- Type of compliance regulations to enforce
- Type of use to support

Ordering process changes

User-based pricing will affect your ordering process. Here are some items related to ordering that may affect you:

- Do you support switching from user-based pricing to processor-based pricing and the reverse?
- How do you handle software license key requirements for products with keyed compliance?
- How do you handle upgrades (increasing the number of authorized users)?
 - Do you use a toll-free number?
 - Do you use the postal system?
 - How do customer records get updated?
 - How is billing initiated?
- How do your customers find out how many users they are entitled to? Do you need to provide some type of documentation that explains how to determine the number of users to order?
- How will you generate the software license keys for your product?
- How will the software license key be delivered to your customer? Some options are mail, fax, phone, or electronic distribution.

License term strategy

The license term indicates whether the authorized usage limit for a product lasts until the next version, next release, or next modification level of your product. The authorized usage limit can be valid for an entire version, for one release, or for one modification level. The license term you define for your product should be based on your pricing strategy. For example, if your customers must purchase each version of your product, and the releases within the version are free, your license term would be by version. Each time a new license term is installed for a product, the customer must set the authorized usage limit.

The following example is for a product licensed by version:

Product 8CE6D11 Release V1R2M0 is licensed by version; its license term is V1. The Work with License Information (WRKLCINF) command is used to set its authorized usage limit to 25. Product 8CE6D11 Release V1R2M5 is installed and

replaces V1R2M0. The usage limit of 25 is still in effect because Release V1R2M5 is within the V1 license term. Product 8CE6D11 Release V2R1M0 is installed and replaces Release V1R2M5. Release V2R1M0 is not within the V1 license term, so the authorized usage limit must be set using the WRKLCINF command for the new V2 license term.

Your release name is in the format of *VxRyMz*. Your licensing strategy must be considered when naming the releases of your product because the license term is derived from the release name:

Vx stands for products licensed by version.

VxRy stands for products licensed by release.

VxRyMz stands for products licensed by modification.

Type of compliance regulations to enforce

The type of compliance determines the action that the product must take when the authorized usage limit is reached or exceeded.

The types of compliance supported are:

- **Warning**

A user is never denied access to a product. When the usage limit is exceeded, a message is sent to the product and the system operator indicating the authorized usage limit is exceeded.

- **Operator action**

A new user is denied access to the product once the usage limit is reached, but users that are currently using the product still have access. The new user cannot access the product until the usage limit is increased by the system operator or a license user is released.

- **Keyed**

License keys are required for products or features using keyed compliance. The license key sets the usage limit and expiration date. Requests for licenses over the usage limit are denied until a new license key is received from the software supplier. After the expiration date is reached, the default usage limit is in effect. When a request for a license is received after the usage limit has been reached, the system sends a warning message to the system operator message queue and to the message queues specified on the Change License Information display. The warning message indicates why the user was or was not granted access to the product.

A grace period can be defined by the product owner to allow the usage limit to be exceeded by 50% for the duration of the period. The grace period begins when a product first exceeds its usage limit. The grace period is the number of days that you have to obtain a new license key. If a new license key is not obtained from the software supplier by the time the grace period has expired, no users over the usage limit are allowed to access the product or feature.

Usage types

The OS/400 license management function monitors use by tracking the number of license users. The following usage types are supported:

- **Concurrent**

This usage limit is for the number of unique jobs using the product at the same time. The license user is the job from which the license request is made.

Registered

This usage limit is for the number of unique license users registered to use the product. A license user is named by the product when the use of a license is requested.

Processor

The usage type is processor, meaning the usage limit is for the number of processors on system partitions where this product or feature is in use. Processor usage type is not supported on the ADDPRDLICI command. Processor usage type is supported in iSeries Navigator and with the application programming interface.

Release considerations

Be aware of the following release-to-release considerations:

- The compliance type, usage type, and the allow license release values specified on the ADDPRDLICI command must be the same for each level of a product within a license term.
- Registered license users are not carried over from one license term to the next. When a new license term is installed, the license users must be registered again.
- Each time a new release of your product is developed, you must test the installation of the product.

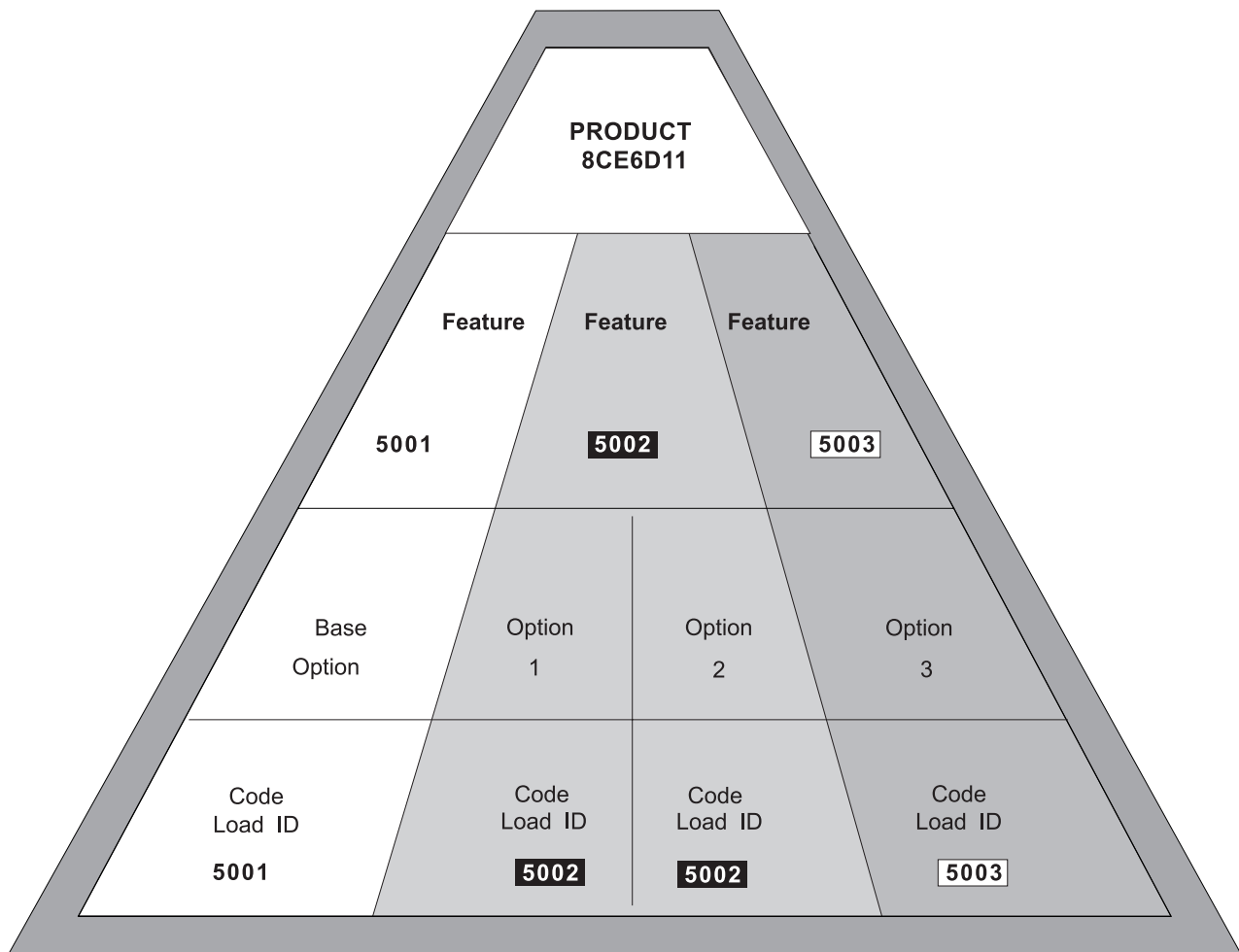
Packaging user-based priced products

User-based priced products must be packaged. For an understanding of packaging products, make sure you read Chapter 3, “Packaging your products” on page 47, and Chapter 2, “Product packaging concepts” on page 37.

Defining features

In order to use OS/400 license management, your product must have a feature number or feature numbers associated with it. A feature is defined as a separately licensed piece of a product that may contain one option or a number of options. All options within a feature have the same code load ID.

Figure 16 on page 97 represents the basic structure of a typical product:



RV2N433-2

Figure 16. Defining Features Using the CRTPRDDFN and CRTPRDLOD Commands

For example, a company, Just Compilers, Incorporated, wants to use user-based pricing on its new compiler product. The company has divided the product into multiple features. The application consists of the following:

Product	Feature	Option	Option Description
8CE6D11	5001	Base	Compiler/400
8CE6D11	5002	1	S/36-compatible compiler
8CE6D11	5002	2	S/38-compatible compiler
8CE6D11	5003	3	*PRV base compiler support

Note: The description for the base option is used on the Work with License Information (WRKLCINF) display to describe each feature.

The following commands define the details about each option and feature for product 8CE6D11 to your system:

```

/*****/
/* Create the product definition to include unique */
/* feature numbers for each option or group of options */
/* that are licensed separately */
/*****/

CRTPRDDFN PRDDFN(COMPILER/COMPILER)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          RGSID(*PHONE 0005075554011)
          MSGF(CMPLRMSG)
          PRDOPT((*BASE MSG0000 *NODYNNAM *NONE 5001)
                (1 MSG0001 *NODYNNAM *BASEOPT 5002)
                (2 MSG0002 *NODYNNAM *BASEOPT 5002)
                (3 MSG0003 *NODYNNAM *BASEOPT 5003)
          CPYRGTFST(*CURRENT)

/*****/
/* Create the product load for each option. The */
/* feature number is used to group options of the */
/* product by the code load ID specified on the */
/* CRTPRDDFN command. */
/*****/

CRTPRDL0D PRDL0D(COMPILERLD)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          OPTION(*BASE)
          LODTYPE(*CODE)
          LODID(5001)
          RGSID(*PRDDFN)
          DVLLIB(COMPILER)

CRTPRDL0D PRDL0D(COMPILER36)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          OPTION(1)
          LODTYPE(*CODE)
          LODID(5002)
          RGSID(*PRDDFN)
          DVLLIB(COMPILER36)

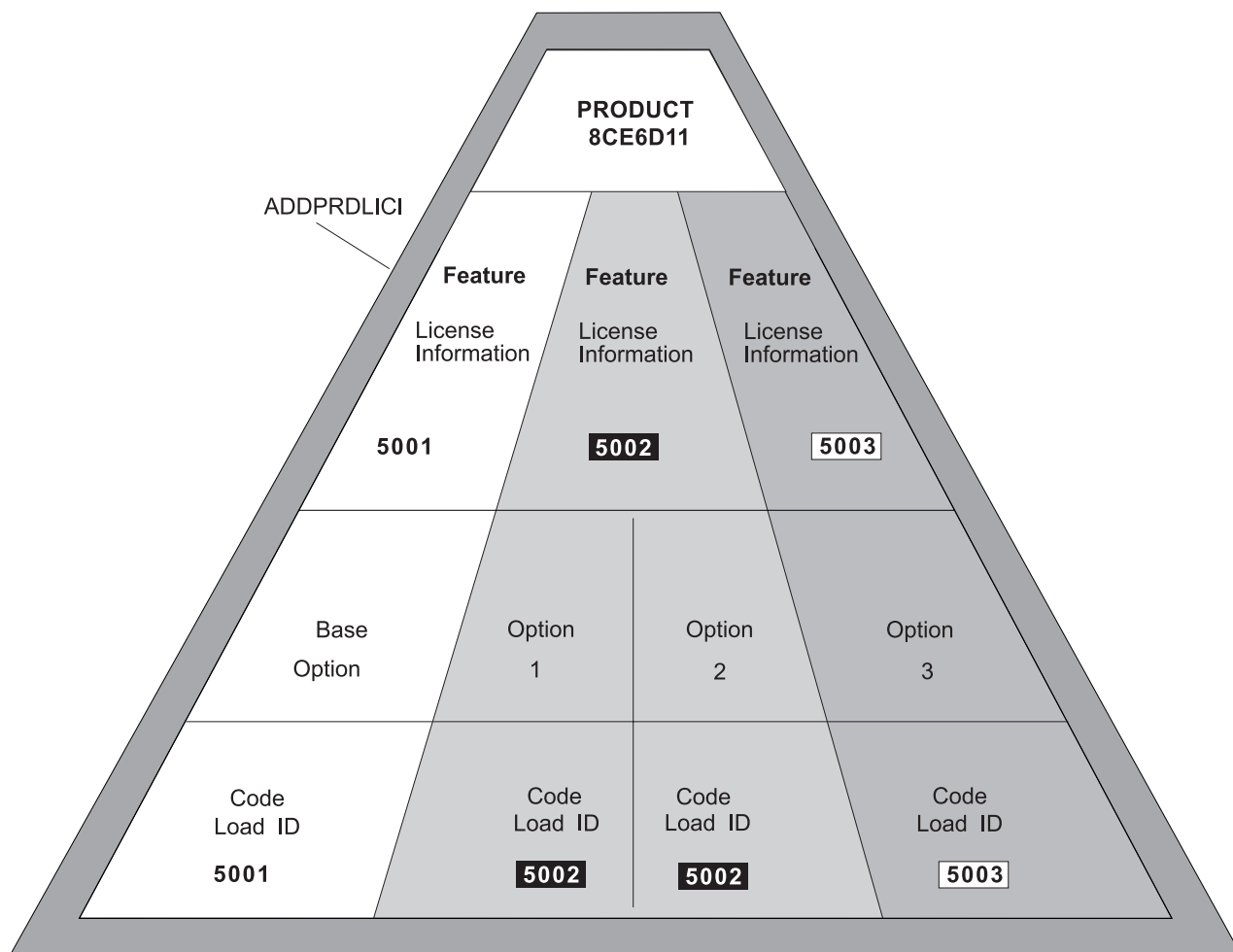
CRTPRDL0D PRDL0D(COMPILER38)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          OPTION(2)
          LODTYPE(*CODE)
          LODID(5002)
          RGSID(*PRDDFN)
          DVLLIB(COMPILER38)

CRTPRDL0D PRDL0D(COMPILERPR)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          OPTION(3)
          LODTYPE(*CODE)
          LODID(5003)
          RGSID(*PRDDFN)
          DVLLIB(COMPILERPR)

```

Adding license information

Once the features for your product have been defined, license information can be added. Figure 17 on page 99 shows the structure of the product after the license information has been added.



RV2N432-5

Figure 17. Adding License Information

You can add license information to IBM products using the Add Product License Information (ADDPRDLICI) command. This command can be used before or after the product or options that belong to a feature are packaged. Use the Package Product Option (PKGPRDOPT) command to package the product or options before they are installed. When you use this command, the following display is shown:

Add Product License Info (ADDPRDLICI)

Type choices, press Enter.

Product ID	_____	Identifier
Release level	_____	VxRxMx
Feature	5001_____	5001-9999
Usage type	*REGISTERED	*CONCURRENT, *REGISTERED
Compliance type	*KEYED _____	*OPRACTION, *WARNING, *KEYED
Default usage limit	1_____	0-999999, *NOMAX
License term	*VERSION_____	*VERSION
Feature Message ID	*BASEOPT_____	Identifier, *BASEOPT
Allow license release	*NO_____	*NO, *YES
Vendor password	*NONE	Name, *NONE
Grace period		
Number of days	0_____	0-999
Allow for default usage limit . .	*NO_____	*NO, *YES

Bottom

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

The commands used in the following example add license information to the product for each feature (the commands for defining the features are also included):

```

/*****/
/* Create the product definition to include unique */
/* feature numbers for each option or group of options */
/* that are licensed separately */
/*****/

CRTPRDDFN PRDDFN(COMPILER/COMPILER)
PRDID(8CE6D11)
RLS(V4R2M0)
RGSID(*PHONE 0005075554011)
MSGF(CMPLRMSG)
PRDOPT((*BASE MSG0000 *NODYNNAM *NONE 5001)
(1 MSG0001 *NODYNNAM *BASEOPT 5002)
(2 MSG0002 *NODYNNAM *BASEOPT 5002)
(3 MSG0003 *NODYNNAM *BASEOPT 5003)
CPYRGTFST(*CURRENT)

/*****/
/* Provide license information for each feature. */
/*****/

ADDPRDLICI PRDID(8CE6D11)
FEATURE(5001)
USGTYPE(*REGISTERED)
COMPLIANCE(*OPRACTION)
DFTUSGLMT(1)
LICTRM(*VERSION)
RLS(V4R2M0)

ADDPRDLICI PRDID(8CE6D11)
FEATURE(5002)
USGTYPE(*REGISTERED)
COMPLIANCE(*OPRACTION)
DFTUSGLMT(1)
LICTRM(*VERSION)
RLS(V4R2M0)

ADDPRDLICI PRDID(8CE6D11)
FEATURE(5003)
USGTYPE(*REGISTERED)

```

```

COMPLIANCE(*OPRACTION)
DFTUSGLMT(1)
LICTRM(*VERSION)
RLS(V4R2M0)

/*****
/* Create the product load for each option.      */
/*****

CRTPRDLOD PRDLOD(COMPILERLD)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          OPTION(*BASE)
          LODTYPE(*CODE)
          LODID(5001)
          RGSID(*PRDDFN)
          DVLLIB(COMPILER)

CRTPRDLOD PRDLOD(COMPILER36)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          OPTION(1)
          LODTYPE(*CODE)
          LODID(5002)
          RGSID(*PRDDFN)
          DVLLIB(COMPILER36)

CRTPRDLOD PRDLOD(COMPILER38)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          OPTION(2)
          LODTYPE(*CODE)
          LODID(5002)
          RGSID(*PRDDFN)
          DVLLIB(COMPILER38)

CRTPRDLOD PRDLOD(COMPILERPR)
          PRDID(8CE6D11)
          RLS(V4R2M0)
          OPTION(3)
          LODTYPE(*CODE)
          LODID(5003)
          RGSID(*PRDDFN)
          DVLLIB(COMPILERPR)

```

The product definition must exist before license information can be added.

License information options

Licenses are stored in internal usage objects or in the license repository. The license repository is used as a central place on the system where licenses are stored.

You can use the following to change, view, retrieve, or remove license information:

CHGLICINF

Change License Information command updates the license information.

DSPLICKEY

Display License Key Information command displays the license information for a product.

QLZARTV

Retrieve License Information API finds license information for a given product, license term, and feature.

RMVLICKEY

Remove License Key Information command removes licenses from the repository.

SNDLIC

The Send License (SNDLIC) command provides the capability to send one or more license keys from the central site system to one or more managed systems. Only the license keys matching the system serial number will be added to the managed system license repository.

Creating a license key

Only those products that use keyed compliance require a software license key. To use the product, a valid license key for each product using keyed compliance must be loaded on the system.

The Generate License Key (GENLICKEY) command creates a license key to allow users to access a product or an option of a product. The license key is specific to the product and system information used with this command. The license key is a combination of 18 characters and numbers (A-F and 0-9). The keys are saved in the repository to keep a history of all keys created. When you enter this command, the following display is shown:

```
Generate License Key (GENLICKEY)

Type choices, press Enter.

Product ID . . . . . _____ Identifier
License term . . . . . _____ Vx, VxRx, RxRxMy
Feature . . . . . _____ 5001-9999
Vendor password . . . . . _____ Name, *NONE
System serial number . . . . . _____ Number
Processor group . . . . . *ANY _____ Number, *ANY
Expiration date . . . . . *NONE _____ Date, *NONE
Usage limit . . . . . 1 _____ 0-999999, *NOMAX
Vendor data . . . . . *NONE _____ Character value, *NONE
Output . . . . . * _____ *, *PRINT, *LICKEYFILE

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

Application program interface (API) calls

In order for OS/400 license management to track the use of your product, you must add application program interface (API) calls to your product to tell the license management function that your product is being used.

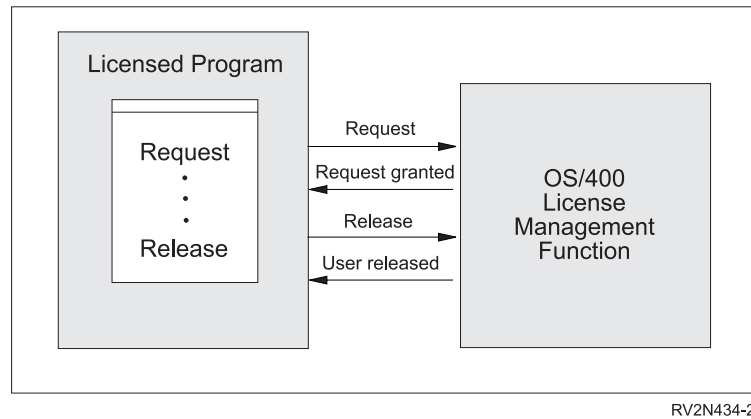


Figure 18. Application Program Interface (API) Calls

- The Request License (QLZAREQ) API is used to indicate that your product is in use by a particular user.
- The Release License (QLZARLS) API is used to indicate your product is no longer in use by a particular user.
- The Retrieve License Information (QLZARTV) API is used to retrieve license information for your product.

Requesting a use of the license

You must determine at which point or points your product has been used. At each of these points, you request a use of the license. For example, you may:

- Request a use of the license in each command processing program.
- Request a use of the license in a single program if all the uses of your product go through one central program.

When your program requests a use of the license, OS/400 license management assigns that use to a license user. The usage type determines how the license user is assigned. For concurrent use, OS/400 license management uses the job qualifier from the program making the request as the license user. For registered use, the license user is defined by the product and passed as an input parameter. Option 8 from the Work with License Information (WRKLCINF) display can be used to display all license users for a product.

Be aware that additional attempts to request the license for a given user (beyond the first) are ignored and no error is signaled.

The sections named “Concurrent use example” on page 105 and “Registered use example” on page 107 further explain concurrent and registered usage.

Releasing a use of the license

A use of the license is assigned to a license user until it is released. To release a use of the license, use the Release License (QLZARLS) API and specify the same parameters that were used on the Request License (QLZAREQ) API.

Be aware of the following considerations when releasing a use of the license:

- Only one release license request is needed for a given user, even if there were several license requests for the same user.

- Additional attempts to release the license for a given user (beyond the first) are ignored and no error is signaled.
- There is no difference (to OS/400 license management) between the following:
 - An attempt to release a license user that already was released
 - An attempt to release a license user that never was requested

The following situations apply to concurrent use only:

- When the job associated with the license user ends, the license management function releases that use of the license from the license user and decreases the usage count. Although the release is done automatically, it is best for the requesting program to do the release as soon as the use of the product ends. This maximizes the number of license uses available at any given point in time.
- If the job is ended abnormally by using the End Job Abnormal (ENDJOBABN) command, that use of the license will not be released automatically and the usage count will not be decreased until the next initial program load (IPL).

The following situation applies to registered use only:

- An option is provided on the Work with License Information (WRKLICINF) display to release registered users. The ALWLICRLS parameter on the Add Product License Information (ADDPRDLICI) command is used to indicate whether this function should be available for your product.

Compliance considerations

Compliance is enforced by the program requesting the use of the license, not by the license management function. License management keeps track of only the current license users and tells the requesting program and the system administrator when the usage limit has been reached or exceeded.

For the Operator action compliance type, the product should prevent access to the product if message CPF9E18 is signaled by the QLZAREQ API when a use of the license is requested. Message CPF9E18 is signaled and the new license user is not added to the list of license users when a product requests a use of the license, and the request would cause the authorized usage limit to be exceeded. The product should not allow new users to access the product until a use of the license is released, or the authorized usage limit is increased.

For the Warning compliance type, the product should allow new users to access the product when the authorized usage limit has been reached. When a product requests a use of the license that will cause the authorized usage limit to be exceeded, OS/400 license management will add the new user to the list of license users and signal message CPF9E17.

For the Keyed compliance type, the product should prevent access to the product if any of the following messages are returned:

- CPF9E11
- CPF9E18
- CPF9E71
- CPF9E73

When these messages are issued, the customer needs to request and load a new license key to increase the usage limit.

All of the messages listed above are also sent to the system operator's message queue (QSYSOPR) and to the queues specified by the system administrator on the Change License Information (CHGLICINF) command.

Authority considerations

All of the license management APIs are shipped with PUBLIC(*EXCLUDE) authority. The product should include a user profile that has private authority to the APIs. This ensures that the request and release of licenses will be successful regardless of the authority of the user running the product. You can have the installation exit program for the product create the user profile and grant the product profile *USE authority to the license management APIs. The program that calls the APIs can then adopt the product's user profile.

See Chapter 2, "Product packaging concepts" on page 37 for more information about installation exit programs.

Requesting a use of the license examples

The following examples use the COMPILER product that was created using the information contained in "Packaging user-based priced products" on page 96. Each example begins with the Work with License Users display for the particular usage type being explained. The examples then demonstrate how the use of a license is requested and released for the COMPILER product, and how the request and release affects the list of license users for the product.

Concurrent use example

For concurrent use, OS/400 license management uses the job qualifier from the program making the request as the license user.

The Work with License Users display shows a list of all users for the Compiler/400 product.

```

Work with License Users

Product ID . . . . . : 8CE6D11
License term . . . . . : V1
Feature . . . . . : 5001
Description . . . . . : Compiler/400
Usage type . . . . . : *CONCURRENT

Type options, press Enter.
  5=Work with job

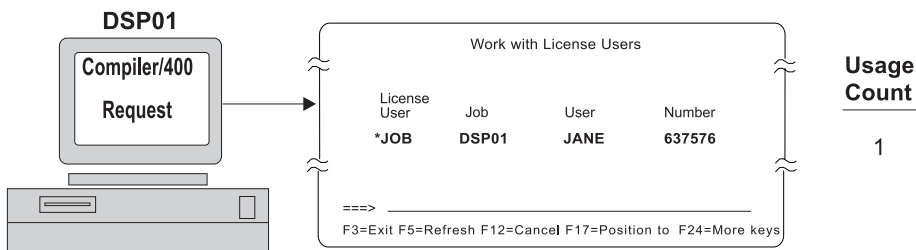
  License      -----Request-----
Opt  User      Date      Time      Job      User      Number
-    *JOB      06/23/94   11:35:47  DSP01    JANE      637576
-    *JOB      06/23/94   12:24:08  DSP04    ERIC      512345
-    *JOB      06/23/94   10:37:51  DSP05    JANE      934545
-    *JOB      06/23/94   11:48:13  WS001    JOHNSON   254345
-    *JOB      06/23/94   10:49:07  DSP03    KPS       353445
-    *JOB      06/23/94   12:35:56  DSP02    RJONES    568885
-    *JOB      06/23/94   11:55:56  DSP04    RJONES    568887
                                           More...

Parameters or command
====>
F3=Exit  F5=Refresh  F12=Cancel  F17=Position to
F21=Nondisplay data/keys

```

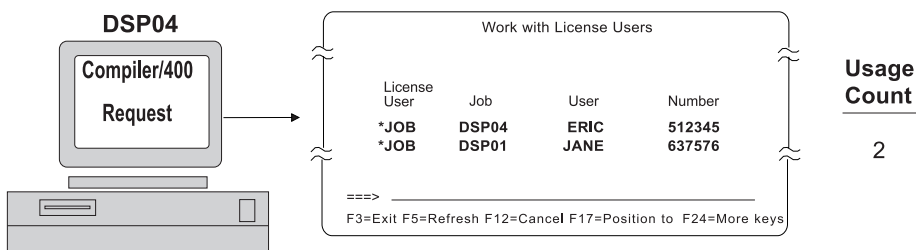
In the following example, the COMPILER/400 product is licensed for concurrent use.

- User JANE compiles a program using COMPILER/400 from workstation DSP01.
 - Compiler/400 uses the Request License (QLZAREQ) API and specifies *JOB as the license user.
 - The job qualifier is assigned as the license user.
 - The usage count is increased from 0 to 1.



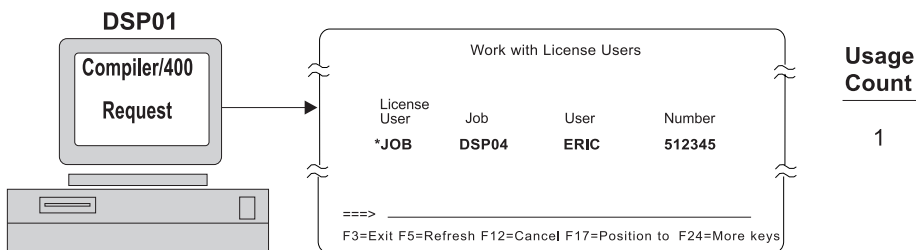
RV3N440-1

- Later, ERIC at DSP04 also compiles a program using COMPILER/400.
 - COMPILER/400 uses the Request License (QLZAREQ) API and specifies *JOB as the license user.
 - The job qualifier for ERIC is assigned as the license user.
 - The usage count is increased from 1 to 2.



RV3N430-1

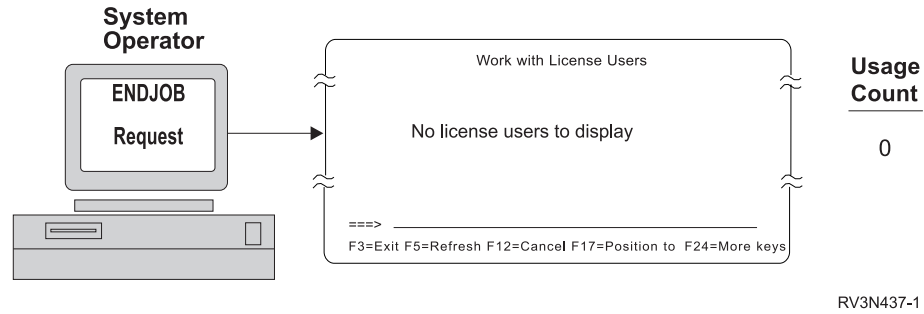
- JANE completes her compilation.
 - COMPILER/400 uses the Release License (QLZARLS) API.
 - The license user associated with JANE is released, so the only user listed on the display is ERIC.
 - The usage count is decreased from 2 to 1.



RV3N436-1

- The system operator cancels job DSP04 ERIC 512345.
 - The license user associated with ERIC is released by OS/400 license management.

- The usage count is decreased from 1 to 0.



Registered use example

For registered use, the license user is defined by the product and passed as an input parameter (for example, a user profile or a device name) using the QLZAREQ API.

The Work with License Users display shows a list of all users for the COMPILER/400 product if the product was packaged for registered use.

```

Work with License Users

Product ID . . . . . : 8CE6D11
License term . . . . . : V1
Feature. . . . . : 5001
Description . . . . . : COMPILER/400
Usage type . . . . . : *REGISTERED

Type options, press Enter.
  5=Display Detail  6=Release License

License      -----Request-----
Opt  User      Date      Time      User
-    ERIC      05/01/93   02:55:56   ERIC
-    JANE      04/30/93   11:35:47   JANE
-    KEVIN     06/28/93   07:26:51   KEVIN
-    SANDY     05/09/93   10:15:07   SANDY
-    TERESA    07/28/93   09:43:32   TERESA

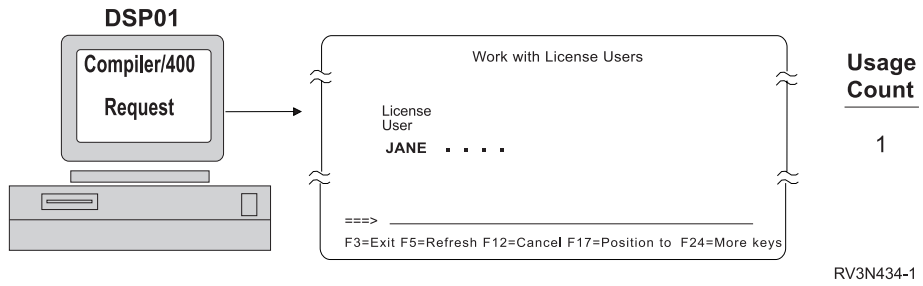
Parameters or command
====>
F3=Exit  F5=Refresh  F12=Cancel  F15=Sort by date/time  F16=Release all
F17=Position to  F21=Nondisplay data/keys

Bottom

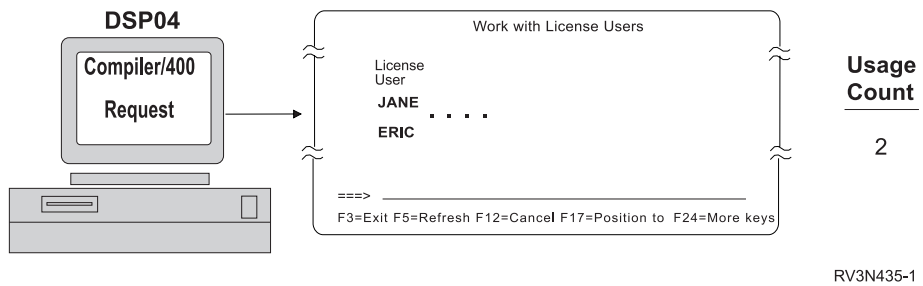
```

In the following example, the COMPILER/400 product is licensed for registered use.

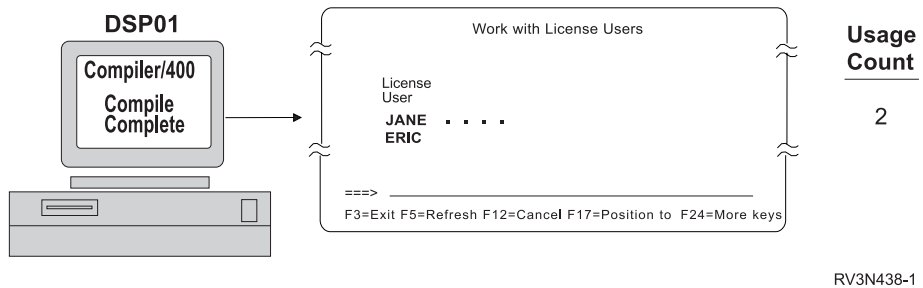
- User JANE compiles a program using COMPILER/400 from workstation DSP01.
 - Compiler/400 uses the Request License (QLZAREQ) API and specifies JANE as the license user.
 - JANE is assigned as the license user.
 - The usage count is increased from 0 to 1.



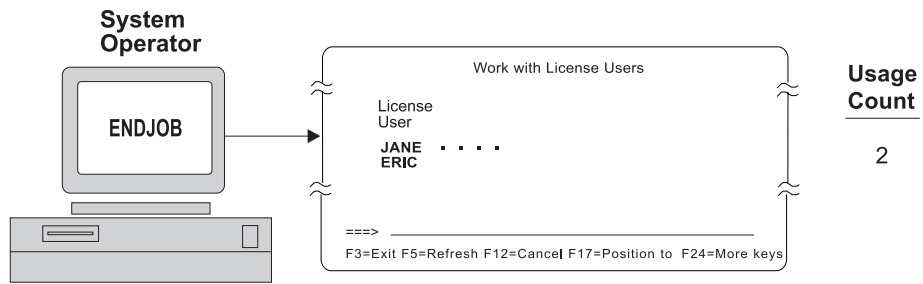
- Later, ERIC at DSP04 also compiles a program using COMPILER/400.
 - Compiler/400 uses the Request License (QLZAREQ) API and specifies ERIC as the license user.
 - ERIC is assigned as the license user.
 - The usage count is increased from 1 to 2.



- JANE'S compile is completed.
 - Since COMPILER/400 has a usage type of registered, a release is not issued and JANE remains as a license user.
 - The usage count remains at 2.



- The system operator cancels ERIC'S job.
 - Since COMPILER/400 has a usage type of registered, ERIC remains as a license user.
 - The usage count remains at 2.



RV3N439-1

Chapter 5. Product installation, distribution, and management

This chapter explains how to manage a product after it has been packaged. The management functions addressed in this chapter include:

- Installing a packaged product
- Avoiding library name conflicts at install time
- Testing an installed product
- Distributing a product
- Considerations for product backup and recovery
- Problem determination for an installed product
- Using product exit programs to manage your product

For the most part, packaged products can be managed just like IBM licensed programs. The following OS/400 commands are used to manage products:

- Analyze Problem (ANZPRB) - allows problem analysis and problem reporting
- Check Product Option (CHKPRDOPT) - determines why the product is not properly installed
- Delete Licensed Program (DLTLICPGM) - deletes the product
- Display Software Resources (DSPSFWRSC) - lists all installed products
- Restore Licensed Program (RSTLICPGM) - restores the product
- Save Licensed Program (SAVLICPGM) - saves the product

Note that the LICPGM menu (GO LICPGM function) operates only on IBM licensed programs.

Note: Installation and management of most products requires a set of instructions. Product owners must provide these instructions with their products. Use the material in this chapter and Chapter 7, "Servicing your products using System Manager" to develop your instructions.

Saving licensed programs

The Save Licensed Program (SAVLICPGM) command saves objects on a list that was created and stored with the product information when the Package Product Option (PKGPRDOPT) command was run. If your product is deleting or renaming any of these objects, use the exit program API and its missing object list. The SAVLICPGM and Check Product Option (CHKPRDOPT) commands call your exit program and pass it a list of missing objects. Your exit program then determines whether it is acceptable for these objects to be missing.

Using the SAVLICPGM command parameters, you can save the following types of products to a save file:

- Your products (after you use the System Manager program to package them)
- Licensed programs
- Licensed Program Offerings (LPOs)
- Programming Requests for Price Quotation (PRPQs)

Installing a packaged product

The following is an overview of what happens when your product is installed by the Restore Licensed Program (RSTLICPGM) command.

1. Product definition, product load objects, and preoperation exit programs are restored into library QTEMP
2. Preoperation exit programs in QTEMP are called and run
3. Product objects are restored into the product library
4. Postoperation exit program is called
5. Additional libraries are processed one at a time:
 - Preoperation exit programs in QTEMP are called and run
 - Product objects are restored into the additional library
 - Postoperation exit program is called

Using the RSTLICPGM command parameters, you can restore or load a licensed program, either for an initial installation or a new-release installation.

Avoiding library name conflicts at installation time

In order for several products to have the same library on a library list for the product load, the products must:

- Have the same registration ID
- Have the same packaged secondary language library name if restoring a secondary language
- Have a primary library name that is the same as the primary library name being restored if restoring code or a primary language

You must also consider language objects when installing code objects. Primary language objects must be installed with code objects. Secondary language objects cannot be installed with code objects.

Note: The list of libraries in a product load for non-IBM products should not use library names that start with "Q". If a library for a non-IBM product starts with "Q" and has the same name as an IBM licensed product, the IBM product will be restored in the library without checking if another product is in the library. If you must put objects into a library that begin with "Q" (for example QS36F), then the objects can be moved or copied by your exit programs.

Testing an installed product

To make sure your product is properly tested, review your Create Product Definition (CRTPRDDFN) command and your Create Product Load (CRTPRDL0D) command for any special attributes that were set. Make sure you test these attributes. For example, if your product allows dynamic naming and multiple releases, you should test that your product functions correctly when these attributes are used.

Distributing your product

After you have packaged your product as described in Chapter 3, “Packaging your products” on page 47 or Chapter 2, “Product packaging concepts” on page 37, you can distribute the product electronically to multiple systems at the same time. The easiest way to get started is to use the Send Product (SNDPRD) command.

Sending a Product

The Send Product (SNDPRD) command provides the capability to send a product, packaged with the System Manager product packaging support, from the central site system to one or more managed systems. You need to know the product ID value to use this command.

This command packages the product for distribution if it is not already packaged. This step can be done manually by using the Managed System Services PKGPRDDST command.

If the product was packaged for distribution before but *REPLACE is specified for the package product for distribution (PKGPRDDST) parameter on the SNDPRD command, the command packages the product for distribution again. It replaces the previous packaged product.

This command also installs the product if the *YES value is specified for the install parameter.

The alternative to using the Send Product (SNDPRD) command is to create a change request description. This is useful when the distribution of your product needs to be coordinated with other steps (like installing a product). If a change request description is used to distribute a product, the product must first be packaged for distribution using the Package Product for Distribution (PKGPRDDST) command. See Chapter 10, “Controlling operations centrally” on page 197 for more information regarding change request descriptions.

Installing a remote product

The Install Remote Product (INSRMTPRD) command provides the capability to install a product, packaged using the System Manager licensed program, on one or more managed systems.

You will need to provide instructions to your customers explaining any special considerations for installing your product.

Product backup and recovery considerations

If you have a system backup, use the book *Backup and Recovery*. Otherwise, install your product using the Restore Licensed Program (RSTLICPGM) command.

Problem analysis for an installed product

Users of your product will need to know how to determine problems after the product is installed. You must provide any special installation instructions to your customers. To determine which information to provide, see “Analyzing problems” on page 150 for a summary of analyzing problems on an installed system.

Chapter 6. Using the program temporary fix (PTF) functions

This chapter provides information about answering problems and managing the service provider program temporary fix (PTF) database and associated PTF save files. This chapter explains how to:

- Order PTFs
- Use PTF save files
- Receive PTFs
- Distribute PTFs
- Create PTF cumulative packages
- Print cover letters

System Manager and OS/400 licensed programs provide commands and functions that simplify these PTF operations. You can combine the commands in control language programs that run in batch. This helps avoid PTF processing during regular business hours. This chapter introduces these commands that are related to the PTF process.

This chapter does provide some suggestions for many command parameter values. However it does not discuss every parameter value for all the available commands.

For information about loading and applying PTFs or Preventive Service Planning packages, see the Software and licensed program topic under Systems Management in the Information Center.

Overview of change management on an OS/400 service provider

Once you install the System Manager program and you set up your system as a service provider, you can use the change management commands and functions to work with PTF save files, cover letters, and information on your system. This chapter provides an overview of some commands and operations you can use to work with PTFs.

Work with supported products (WRKSPTPRD) command

To indicate if your service requesters will support (supply service for) new products, type WRKSPTPRD at the command line. The Work Supported Products display allows you to indicate if your service provider supports (supplies service) for a new product. Determine your service provider support when service requesters install and purchase new products.

To receive PTFs at your service providers, you must install or support the products that they service. To specify the list of products, product options, and language features that you want to process service request and PTF orders, use the Work with Supported Products (WRKSPTPRD) command.

Recall that you used the WRKSPTPRD command to set up product support in Chapter 1, "Setting Up for System Manager" on page 1. To ensure product support for your network is current, use the WRKSPTPRD command as new licensed programs are installed. For the OS/400 program a service provider may provide support only for certain releases. For information about which releases of the

OS/400 programs and IBM licensed programs a service provider can support, see Table 2 on page 5. You can use the Release (RLS) parameter to specify the release with which you want to work.

Ordering program temporary fixes (PTFs)

PTFs correct problems or potential problems. PTFs also can provide new system function. You can send a service request to order the necessary PTFs from the next level service provider or from IBM service support.

Before you begin ordering PTFs to store as save files on the service provider for distribution, you should order and apply the latest cumulative PTF package. Cumulative PTF packages contain new and previously distributed PTFs. Cumulative PTF packages are updated on a regular basis. Apply any cumulative PTF packages on all the service requesters for which you are providing support. This reduces the number of PTFs you will order and distribute later.

Some PTF orders require special handling when received at an OS/400 service provider from a service requester. The following list identifies these special PTF packages and indicates what they are:

- MF98vrmm - Hardware Service Planning Package
- SF98vrmm - Preventive Service Planning Package
- SF99vrmm - Cumulative PTF Package
- SF99vrmm - PTF group
- SF97vrmm - PTF Summary Package

The vrm identifier stands for the version, release, and modification level of the operating system for which you are ordering the PTF package. You can order these PTF packages individually or as a part of a list. If a service requester orders a cumulative PTF package for a previous release that you support, forward the PTF order to IBM service support. Change the contact information so the cumulative PTF package is mailed to the location of the service requester.

You can order most PTFs as part of a mixed list. A mixed list PTF order can contain operating system PTFs, most licensed program PTFs, and Licensed Internal Code fixes. Only PTFs that begin with the prefix SH, are not included in the mixed list. Order these PTFs separately. Information APARs are coded with the II prefix and are not considered PTFs and must also be ordered separately.

The following sections describe the commands available for ordering PTFs.

Ordering the PTFs that you need

To order PTFs and any accompanying information, type SNDPTFORD at the command line. From this display, you can order the following:

- Separate or accompanying cover letters
- Individual PTFs
- Multiple PTFs
- PTF Summary
- Preventive service planning tips
- Cumulative PTF packages
- PTF groups

You can customize your order to be a combination of any of these options.

Ordering cover letters

You can order a PTF that comes with a cover letter, or you can order the cover letter separately. Both service providers and service requesters can order one or more specific PTFs or cover letters.

To preview the changes a PTF provides before ordering that PTF, order the cover letter. You can order a cover letter individually or in packages separately from the PTFs they describe, specify the *CVRLTR value for the PTF parts parameter on the Send PTF Order (SNDPTFORD) display.

If you order a cover letter for a second time, it replaces the original cover letter if it still exists on the system. If you edit your cover letters before, do not accidentally over-write your changes. Cover letters are sent over the service link only. You will receive messages that confirm your cover letter order.

Note: Unless specifically requested, PTFs that are already installed or ordered will not be ordered again.

Some restrictions to using the SNDPTFORD command are:

- A service requester cannot order a cumulative PTF package from an OS/400 service provider. You can only order PTF packages from your service support.
- A service requester cannot order a PTF group from an OS/400 service provider.
- Only service providers can forward a service request. A service provider cannot forward a PTF order to service support.
- The system checks for prerequisites and corequisites, and the ORDER parameter value is *REQUIRED. See “Checking for prerequisite and corequisite PTFs” on page 122 for more information.

Any system that is designated as a service requester can order PTFs. When ordering PTFs, you can display a list of service providers. The operator at the service requester can select the destination for the order from the list. You can request that PTF to perform interactively, batch, or delayed for later processing. This command also allows you to do the following functions:

- Order a list of PTFs with cover letters
- Define whether or not the requisite PTFs should be sent along with the specific PTFs requested
- Define the service provider that is the target of the request
- Allow use in batch
- Define the mode of delivery as the electronic customer support service link only, or as any other possible method of delivery such as IBM Service (*IBMSRV).

Note: If the PTF is too large to deliver electronically, IBM service support will send it on tape. An OS/400 service provider may not necessarily be able to provide a PTF package on CD-ROM. Other service providers will only be able to provide PTF packages on tape. The network operator and the system operator at the individual service requester works this out.

If you plan to use the Send PTF Order (SNDPTFORD) command in a control language (CL) program to be run in batch, the remote control point name and network ID are required parameters. When they are specified, be sure the control

point name and network ID match the service provider information on your system (Work with Service Provider (WRKSRVPVD) command) or an error will occur.

The following prompt display shows the default parameters for the Send PTF Order (SNDPTFORD) command.

```

Send PTF Order (SNDPTFORD)

Type choices, press Enter.

PTF description:
PTF identifier . . . . .
Product . . . . . *ONLYPRD
Release . . . . . *ONLYRLS
      + for more values
PTF parts . . . . . *ALL
Remote control point . . . . . *IBMSRV
Remote network identifier . . . . *NETATR

Character value
F4 for list
*ONLYRLS, VxRxMx
*ALL, *CVRLTR
Name, *IBMSRV, *SELECT
Name, *NETATR

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Note: The delivery, order, reorder, and check PTF parameters are displayed only when you press the function key provided to show additional parameters.

When you specify *ANY for the delivery method parameter and the PTF order is too large to deliver electronically, a CD-ROM is sent. If there are any problems with the order, your service representative is notified and will contact you. It may be that you need to order and apply the latest cumulative PTF package to bring your system up to date.

When you specify *LINKONLY for the delivery method and the PTF order is too large to deliver electronically, messages are displayed indicating the order is too large. You may want to either break up the original order into smaller orders or order the latest cumulative PTF package.

The reorder parameter allows you to order PTFs that you have already ordered or are already on the system. This is useful for PTFs that have been loaded and applied and for which a save file no longer exists, or for the case of a missing or damaged save file.

The check PTF parameter checks the service requester system to determine which PTFs you ordered. The system determines which products your service requester system supports. You can order PTFs specified on the PTFID parameter even though they are not installed or supported by the service requester system. To order PTFs that are specified on the PTFID parameter, enter *NO in the check PTF parameter field.

Automatically ordering the PTFs that you need from IBM service support

Your system can automatically order the PTFs that it needs. The Order Supported Product PTFs (ORDSPTPTF) function provides “give me what I need” PTF ordering capability for supported products, product options, and language features. This orders and sends even the newest PTFs that are not yet eligible to be in a cumulative PTF package. Before ordering PTFs, be sure that you understand the following:

- To automatically search your system for needed PTFs, type ORDSPTPTF at the command line. This shows the Order Supported Product PTF display.
- If you specify *ANY for the delivery method and the PTF order is too large to be sent electronically, the PTFs are sent on distribution media.
- If you do not want to use media, you can specify *LINKONLY for the delivery mode.
- To limit the number of PTFs ordered, use the licensed program name.
- To avoid waiting a long time for the media, request to receive PTFs electronically. However, instead of specifying *ALL in the product parameter field, specify the product ID for each product that you have installed. Although, you must run ORDSPTPRD several times, you reduce the time it takes to receive your PTFs.

Use the Order Supported Product PTFs (ORDSPTPTF) function command to order PTFs from IBM service support only. This command does not order PTFs from other iSeries servers. This command is only available to OS/400 service providers with System Manager installed.

How the ORDSPTPTF command functions

When the Order Supported Products PTF (ORDSPTPTF) command is processed, the PTFs that are on the service provider, either installed or in save files, are compared with what is available from IBM service support. The naming convention SI95xxx identifies PTFs available from IBM service support. The resulting order is all PTFs for products supported by the service provider that are not already on the service provider or currently on order.

The 120 shows the default parameters for the Order Supported Product PTFs (ORDSPTPTF) command. There is no default value for the Delivery method parameter; you must specify a value for this field.

```

Order Supported Product PTFs (ORDSPTPTF)

Type choices, press Enter.

Delivery method . . . . . *ANY, *LINKONLY
Product . . . . . *ALL Character value, *ALL

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

```

Special PTF notifications

When you order PTFs for one or more licensed programs using the Order Supported Product PTFs (ORDSPTPTF) command, you may receive message notifications indicating that you have a defective PTF on your system or a high-impact pervasive (HIPER) PTF that is missing from or not applied to your system. This notification occurs for defective and HIPER PTFs that are known for all the products that you have installed on the service provider and for all the products you currently support. The messages are sent to the service provider message queue you defined when you set up your system as a service provider using the Change Service Provider Attributes (CHGSRVPVDA) command.

A defective PTF may be marked as such because of a misspelling or grammatical error in the cover letter, or it may not correct the problem, or it may correct the problem and cause a new problem.

If you have a defective PTF on your system for a product you support or have installed, you will receive messages about that PTF when:

- The PTF is on order and the product is supported or installed
- The PTF is in a save file on the system and the product is supported or installed
- The PTF is damaged
- The PTF is loaded
- The PTF is temporarily or permanently applied

The messages notifying you about a defective PTF are received each time you use the Order Supported Product PTFs (ORDSPTPTF) command until you do one of the following:

- Completely remove the PTF from your system and delete the defective PTF save file.
- Apply a superseding PTF when one becomes available and delete the defective PTF.

You will receive messages about missing HIPER PTFs or HIPER PTFs that are on your system but not applied when:

- The PTF is on order and the product is supported or installed

- The PTF is in a save file that is held and the product is supported or installed
- The PTF is not temporarily or permanently applied and the product is installed
- The PTF has not been superseded and the product is installed

When you use the Order Supported Product PTF (ORDSPTPTF) command for a specific product, and a HIPER PTF is available for that product, it is automatically ordered from IBM service support. When you specify *ALL for the Licensed program parameter, all the HIPER PTFs that are available for the products you are supporting are automatically ordered from IBM service support. The messages notifying you about a HIPER PTF will be received each time you use the Order Supported Product PTF command until you do one of the following:

- Temporarily or permanently apply the PTF (when it is available as a save file on your system)
- Temporarily or permanently apply a superseding PTF when one becomes available

Once you have applied either the HIPER PTF or a superseding PTF, you need to release the save file for the HIPER PTF to allow distribution to your service requesters, if you have already distributed it or it has been superseded.

To be sure you receive these messages, you must know to which message queue your service provider messages are being sent. This message queue is the one you defined when you set your service provider attributes. (See “Setting the service provider attributes” on page 18 for the procedure to set or change this message queue.) To see which message queue you are currently using to receive service provider messages, use the Display Service Provider Attributes (DSPSRVPVDA) command.

Restrictions for the Order Supported Product PTFs (ORDSPTPTF) command

If you have limited the ORDSPTPTF command using the parameters and continue to exceed the order limit, you should use the Send PTF Order (SNDPTFORD) command to order the latest cumulative PTF package.

The ORDSPTPTF command does not order prerequisite PTFs outside the product specified. To ensure all the necessary prerequisite PTFs are ordered, use the *ALL value for the licensed program name parameter. If you need to limit the PTF order on a program-by-program basis, be sure to run the command for each licensed program you support. Then use the command with the *ALL value for the licensed program name parameter to order any necessary requisite PTFs that may be out of the scope of an individual licensed program.

The size of PTFs that can be distributed electronically and the number of PTFs that can be sent on distribution media varies from location to location. Check with your IBM Service representative to find out more about restrictions for your location.

PTFs that have been temporarily applied and permanently removed using the Remove Program Temporary Fix (RMVPTF) command are not ordered again when you use the ORDSPTPTF command.

Receiving program temporary fixes (PTFs)

Before you begin to download PTFs from the IBM database or transfer PTFs from the media, you need some background information on how to effectively store PTFs on your system. PTFs are either sent to your system over the electronic customer support service link or received on distribution media, depending on the

size and number of PTFs ordered. If they are sent over the electronic customer support service link, you receive them as save files and they are stored in library QGPL. If the PTFs are received on media, you need to use the Copy PTF to Save File (CPYPTFSAVF) command discussed later in this section to transfer the PTFs into save files and store them in library QGPL. When PTFs are received and transferred to save files in library QGPL through either of these methods, all the PTF identification information and problem symptom data is stored in the PTF database.

Note: If you receive PTFs by any other method, you can use the QPZLOGFX API to log the PTF information. Once the information is recorded or logged, the PTF can be distributed in the usual manner.

Checking for prerequisite and corequisite PTFs

When PTFs are received as a result of a PTF order, prerequisite and corequisite checking is done. This ensures that complete and accurate PTF packages are sent. To check for PTFs, specify ORDER(*REQUIRED) on the Send PTF Order (SNDPTFORD) command. The service provider does the checking. A **prerequisite** PTF is one that must be applied before the PTF that was ordered. A **corequisite** PTF is one that needs one or more PTFs to also be applied along with PTF that you ordered. Corequisite PTFs are sets of PTFs that provide fixes only when they are all on the system. If you order one corequisite PTF, the other is ordered automatically.

IBM Service support (*IBMSRV) defines a third type of requisite PTFs that are called distribution requisites. Distribution requisites are similar to corequisites. They also provide fixes only when they are all on the system. However, the system does not check that all distribution requisites are applied at the same time. You must make sure that they are applied when you apply the PTF that you ordered. All distribution requisites are sent automatically when ordering PTFs from *IBMSRV. However, distribution requisite checking is not done when ordering PTFs between iSeries servers.

Receiving PTFs from IBM service support electronically

PTFs are received by the service provider from IBM service support (*IBMSRV) using advanced program-to-program communications (APPC). A save file is created in library QGPL to contain each PTF received. Cover letters received from IBM service support (*IBMSRV) or from an OS/400 service provider are placed in a physical file member. The library/file name is QGPL/QAPZCOVER. The member name is based on the PTF number or time stamp and may be found in the text of the message displayed when the cover letter files are created. See "Working with cover letters" on page 125 for instructions on displaying and printing the PTF cover letters.

Receiving PTFs from OS/400 service providers electronically

Object distribution, change management services, or Management Central can be used for sending PTFs from an OS/400 service provider to an OS/400 service requester. Object distribution provides the support required by System Manager for distribution of PTFs to other iSeries servers. Distribution can be automatic if the send PTFs automatically (SNDPTFAUTO) service provider attribute is set to *YES. For more information about the service provider attributes, see "Setting the service provider attributes" on page 18

The following file types are sent between iSeries servers for PTF support:

- Save file - contains the PTF
- Physical file member - describes the change and provides instructions (PTF cover letter)

Notes:

1. PTFs can be sent using the *SVDS distribution queue when the service requester has installed the Managed System Services licensed program.
2. If the service provider set the distribution queue to *SNADS in the service requester file, then PTFs are distributed electronically using the SNADS object distribution function. The PTFs are received by a service requester by way of the QSRVBAS user profile within a job called QESTP. QESTP is the job that runs in the QSNADS subsystem that is used to automatically receive PTFs distributed by a service provider. A system cannot receive PTFs when the QSRVBAS message queue is in use by a job other than QESTP.
3. For more information about using Management Central to distribute PTFs, see the Software and licensed program topic under the Systems Management category in the Information Center.

The following section describes the command used to copy PTFs from distribution media to save files for distribution by way of the service link.

Receiving PTFs from distribution media

When you want to electronically distribute PTFs received from physical media, such as CD-ROM or tape, you must copy all PTFs from the media to PTF save files. To copy all PTFs to PTF save files, type CPYPTFSAVF at the command line. The Copy PTF Save File (CPYPTFSAVF) command copies all PTFs associated with supported or installed products from the physical media to PTF save files. The save files are stored and created in library QGPL. The PTF save file is named Qxxxxxxx, where xxxxxxx is the PTF identifier or timestamp. Each PTF will reside in a separate save file just as if it had been received over the electronic customer support service link. If a PTF save file already exists, then that PTF is skipped and a message is sent to the job log indicating the PTF already exists in a save file. No checking for requisite or superseded PTFs is done.

Cover letters on the tape are copied and placed in a member in library/file QGPL/QAPZCOVER. However, PTF summary lists are not copied. If you need a specific cover letter or a summary list that was not found on the tape, use the SNDPTFORD command and specify the *CVRLTR parameter to indicate you want cover letters only.

After all the PTFs you requested are copied into save files, a message is sent to the job log indicating how many PTFs were copied and how many were not. If any were not copied, check the previously listed detailed messages in the job log to determine why these PTFs were not copied.

The following prompt display shows the default values you may use with the CPYPTFSAVF command.

```

                                Copy PTF to Save File (CPYPTFSAVF)

Type choices, press Enter.

From device . . . . . Name
Product . . . . . *ALL      Character value, *ALL...
PTF numbers to select . . . . *ALL      Character value, *ALL
      + for more values
PTF numbers to omit . . . . .      Character value
      + for more values

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

```

A save file is created in QGPL for each PTF copied. If a save file for a PTF already exists, then the PTF is not copied. Messages are sent to the job log and direct you to the Display Program Temporary Fix (DSPPTF) command for information on specific PTFs. The reasons that a PTF would not be copied include:

- The PTF is not found.
- The PTF is for a product that is not supported or installed.
- The PTF already exists

This command does not do requisite checking. If you are going to copy an entire package, it is recommended that you run the CPYPTFSAVF command in a batch environment because of the amount of time it may take to complete.

Working with PTF save files

Whether you receive PTFs electronically as save files or use the Copy PTF to Save File (CPYPTFSAVF) command to copy PTFs from the distribution media (such as a tape or CD-ROM) into save files, all the PTF save files are stored in library QGPL. PTF save files are named Qpppppppp, where pppppppp is the PTF ID or a timestamp. To avoid confusion and problems with distribution, you should **NOT** place files in library QGPL that begin with the letter Q and end with four numbers (these are reserved for PTF save files only).

When you need to back up your system, the save/restore operations, SAVLIB/RSTLIB, backup and restore the PTF save files. The PTF and cover letter information is not saved but is built again when the product is installed. To rebuild the information in the database, you need to delete the QAPZ* files in the QUSRSYS and start the system again. If you have a large number of save files, this IPL could take some time.

When you receive a PTF that incorporates or takes the place of (supersedes) a PTF, or you become aware of a defective PTF, use the Delete Program Temporary Fix (DLTPTF) command to delete both the save file and all the associated PTF information. When you upgrade your system to a new release of the OS/400 program, you may also need to delete the PTFs from the previous release. This way you control the amount of storage space used for PTF management.

Note: If you want to delete a PTF save file, you should always use the Delete Program Temporary Fix (DLTPTF) command.

For more information about the Delete Program Temporary Fix (DLTPTF) command and the procedures to reclaim storage following the deletion of PTFs and PTF information, see “Deleting save files, cover letters, and on-order status” on page 132.

Note: Do not use the following methods to process PTFs that you want to distribute to other servers:

- PTF save files created by the Copy Program Temporary Fix (CPYPTF) command
- Object distribution over a line that is not an electronic customer support link

When using these methods, no PTF information is written into the PTF database. Use the following APIs to store information about the PTFs into the PTF database file.

Log PTF Information (QPZLOGFX)
Generate PTF Name (QPZGENNM)
Retrieve PTF Information (QPZRTVFX)

PTF cover letters

Cover letters provide information about the PTFs that they accompany. You receive cover letters electronically as physical file members or with PTFs on tape. The following sections explain how to do the following:

- Display a cover letter
- Display cover letters related to specific problems
- Printing cover letters

Working with cover letters

Use the Copy PTF to Save File (CPYPTFSAVF) command to copy PTFs and cover letters from the media to individual files. The cover letters are stored in members of file QAPZCOVER in library QGPL and are named Qppppppppxx, where the pppppppp is the PTF ID or a timestamp depending on what the PTF Save File was named. The xx is the last two digits of the language code of the cover letter. For cover letters in English, the last two digits of the language code do not appear and the member name is the same as the corresponding PTF save file name.

The preferred language for cover letters for a service requester is defined in the contact information on the service requester and may only be changed interactively using the Work with Contact Information (WRKCNTINF) command. If you want to order a cover letter in a language that is different from the language used on your system, you need to change the language code in the contact information displayed when you use the Send PTF Order (SNDPTFORD) command.

The Send PTF Order (SNDPTFORD) command allows you to specify whether you want to order complete PTFs or cover letters only (*CVRLTR) for PTF parts). Use the cover letters only option by specifying PTFPART(*CVRLTR) to preview new PTFs and determine whether or not you want to order that PTF.

To preview a cover letter, you need to be able to display or print the cover letter using the following procedures.

Displaying cover letters

To view a PTF cover letter, do one of the following:

- Prompt the Display PTF Cover Letter (DSPPTFCVR) command and enter the appropriate values.
- Prompt the Display PTF (DSPPTF) command and enter the product ID, PTF ID, release, and specify COVERONLY(*YES).

The DSPPTF command only displays cover letters and PTFs for products that are installed or supported.

If the cover letter is available in more than one language, a display is shown that allows you to select the language in which you want to print the cover letter.

- Prompt the Display PTF (DSPPTF) command and specify the product ID. Press F11 on the Display PTF status screen for the second view.

The Cover letter column indicates whether a cover letter exists for a specific PTF. Enter option 8 next to the PTF for which you want to display the cover letter.

- Prompt the Work with PTF (WRKPTF) command and enter the product ID and release of the PTF.

Enter an 8 next to the PTF for which you want to display the cover letter.

If the cover letter is available in more than one language, a display is shown that allows you to select the language in which you want to view the cover letter.

Displaying cover letters for specific problems

To view a PTF cover letter that is related to a specific problem, do the following:

1. Type the Work with Problem (WRKPRB) command on any command line and press the Enter key. The Work with Problems display is shown.

The WRKPRB command gives you the option to display the cover letter.

2. Type a 5 (Display details) in the *Opt* column next to the problem record with which the PTF is associated and press the Enter key.

3. Press F9 (Display PTFs) to display the PTFs.

4. Type an 8 (Display cover letter) in the *Opt* column next to the PTF number for which you want to display the associated cover letter. Press the Enter key.

The cover letter is displayed.

Printing cover letters

To print a PTF cover letter, do one of the following:

- Prompt the Display PTF Cover Letter (DSPPTFCVR) command and enter OUTPUT(*PRINT).
- Prompt the Display PTF (DSPPTF) command and enter the product ID, PTF ID, release, COVERONLY(*YES), and OUTPUT(*PRINT).

If the cover letter is available in more than one language, a display is shown that allows you to select the language in which you want to view the cover letter.

- Prompt the Display PTF (DSPPTF) command and enter the product. Press the Enter key. Enter a 6 next to the PTF (on the Display PTF status display) for which you want to print the cover letter.
- Prompt the Work with PTF (WRKPTF) command and specify product ID and release. Enter option 6 next to the PTF (on the Work with PTF status display) for which you want to print the cover letter.

Note: You can print all the cover letters that are specified for the PTF. This is only possible, if the Display PTF (DSPPTF) command is run in batch and the cover letter is available in more than one language.

Work with PTF commands

The following commands are provided to allow you to work with PTFs on your system.

Work with Program Temporary Fixes (WRKPTF) Command

The Work with Program Temporary Fixes (WRKPTF) command allows you to select and work with all the PTFs associated with a specified product and release level.

The Work with PTF menu helps you to:

- Copy, create, delete, hold, remove, or release a PTF
- Display the PTF cover letter
- Display details of the PTF
- Load and apply the PTF
- Print the cover letter
- Send the PTF
- Work with problems
- Assign a PTF range
- Work with assigned PTF IDs

```
                                Work with PTF
                                System:  XXXXXXXX
Product ID . . . . . : 57XSS1
Release of base option . . . . . : V4R2M0 L00

Type options, press Enter. To work with assigned PTF IDs, press F18.
 1=Create   3=Hold   4=Delete  5=Display details  6=Release
 9=Work with problems  11=Load/Apply ...

Opt PTF      Status                               PTF save file

(No PTFs available)

Parameters or command
===>
F3=Exit  F11=Display option  F12=Cancel  F17=Position to
F18=Work with assigned PTF IDs  F23=More options  F24=More keys

Bottom
```

The associated function keys on the Work with PTF menu help you to order supported product PTFs and display a PTF's status and options.

Working with assigned PTF IDs

The Work with Assigned PTF IDs display shows PTF IDs that have been assigned to problems, but the PTF save file is not known to the system.

The Work with Assigned PTF IDs menu helps you to :

- Create a PTF for the assigned PTF ID
- Work with problems associated with the PTF ID

To work with assigned PTF IDs:

1. Prompt the Work with PTF (WRKPTF) command and enter the product ID and the release.

2. Press F18 to work with assigned PTF IDs.

```
Work with Assigned PTF IDs
System: XXXXXXXX
Product ID . . . . . : 1X12345
Release . . . . . : V1R1M0

These PTF IDs have been assigned to problems, but the PTFs have not been
created.

Type options, press Enter.
1=Create PTF 9=Work with problems
Opt   PTF ID
      1ZT0000

Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel  F17=Position to
Bottom
```

Change the PTF range

The Change PTF Range display allows you to display or change the range of PTF IDs for a specified product and release. This is the range of PTFs that are valid for the Create PTF (CRTPTF) command.

To Change the PTF Range:

1. Prompt the Work with PTF (WRKPTF) command and enter the product ID and the release.
2. Press F19 to change the PTF range.

```
Change PTF Range
Product ID . . . . . : 1X12345
Release . . . . . : V1R1M0

Type changes, press Enter.

Prefix . . . . .
Lowest number . . . . .
Highest number . . . . .
Next Available . . . . : None

F3=Exit  F5=Refresh  F12=Cancel
```

Display Program Temporary Fix (DSPPTF) command

The Display Program Temporary Fix (DSPPTF) command allows you to display or create an output file of information about all the PTFs on your system. This command only shows the PTFs for installed or supported products and is available

to both the service provider and service requester. Use this command to help you manage the PTFs that are on order, and the PTFs that you receive for change management in your network.

Note: When you use the Display Program Temporary Fix (DSPPTF) command, PTFs sent electronically from IBM service support, cover letter orders, and cumulative PTF orders never appear as “on order”.

The following prompt display shows the default parameter values for the Display Program Temporary Fix (DSPPTF) command when you choose to use the *OUTFILE parameter.

```

Display Program Temporary Fix (DSPPTF)

Type choices, press Enter.

Product . . . . . *ALL          F4 for list
PTF numbers to select . . . . . *ALL      Character value, *ALL...
Release . . . . . > *ALL          *ALL, VxRxMx
Cover letter only . . . . . > *NO       *NO, *YES
Output . . . . . > *OUTFILE        *, *PRINT, *OUTFILE
File to receive output . . . . .        Name
Library . . . . . *LIBL           Name, *LIBL, *CURLIB
Output member options:
Member to receive output . . . *FIRST   Name, *FIRST
Replace or add records . . . *REPLACE  *REPLACE, *ADD

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

```

Example that uses the Display PTF (DSPPTF) command and IBM Query

The following example uses the OUTFILE support available with the Display Program Temporary Fix (DSPPTF) command along with the IBM Query licensed program to produce a file containing an overview of the PTFs on your system and their status. The file can be displayed or printed so you can use it as part of a report.

The first step collects all the PTF information in one output file. The second step creates a query with selected fields. The third step runs the query to place the requested information in a formatted file.

Collect PTF information

1. Type the Create library (CRTLIB) command on any command line.
`CRTLIB LIB(PTFINF)`
2. Type the Display PTF (DSPPTF) command with the following parameters on any command line.
`DSPPTF LICPGM(*ALL) SELECT(*ALL) OUTPUT(*OUTFILE)
OUTFILE(PTFINF/PTFLIST)`

The following message is shown:

Output file PTFLIST created in library PTFINF.

Create the query

1. Type the Start Query (STRQRY) command on any command line.
2. Select option 1 on the Queries Utility display.
3. Type a 1 in the option field, the name you want to call the query, and the name of the library you created for PTF information.

```

Option 1 (=create)

Query..... LIST
Library..... PTFINF

```

4. Press the Enter key. The Define the Query display is shown.
5. Type a 1 (Select) in the *Opt* column next to the Specify file selections query definition option and press the Enter key. The Specify File Selections display is shown.
6. Type the name of the output file you created with the Display PTF command and the name of the library you created for PTF information.

```

File..... PTFLIST
Library.. PTFINF
Member... *FIRST

```

7. Press the Enter key **twice**.
8. Type a 1 (Select) in the *Opt* column next to the Select and sequence fields query definition option and press the Enter key.

The following displays show the possible field descriptions. Press F11 (Display text) to show the possible field names.

Select and Sequence Fields

Type sequence number (0-9999) for the names of up to 500 fields to appear in the report, press Enter.

Seq	Field	Seq	Field	Seq	Field
	SCCENTRY		SCSTATUS		SCSPTF
	SCDATE		SCSTDATE		SCENST
	SCTIME		SCSTTIME		SCENDT
	SCSRLNBR		SCTYPE		SCENTP
	SCXPFID		SCIPLACT		SCENIA
	SCXPFREL		SCLANG		SCENLG
	SCXPFMOD		SCOBJLIB		SCXPFV
	SCXPFDAT		SCISRC		SCPTFV
	SCLIB		SCSYSNM		SCMCGG
	SCPPID		SCONOR		SCACTP
	SCPTFID		SCSAVF		SCEACP
	SCPPREL		SCOPTP		SCAACR

More...

F3=Exit	F5=Report	F11=Display text	F12=Cancel
F13=Layout	F20=Renumber	F21=Select all	F24=More keys

When you press the Page Down key, more fields are displayed.

Select and Sequence Fields

Type sequence number (0-9999) for the names of up to 500 fields to appear in the report, press Enter.

Seq	Field	Seq	Field	Seq	Field
	SCEACR				SCTGTV

Bottom

F3=Exit	F5=Report	F11=Display text	F12=Cancel
F13=Layout	F20=Renumber	F21=Select all	F24=More keys

9. In this example, the following fields with the numbers shown were used:

1	SCONOR	On Order
2	SCPTFID	PTF
3	SCPPID	Licensed Program
4	SCPTFV	PTF Release
5	SCOBJLIB	Object Library
6	SCSTATUS	Status

Press the Enter key twice to process and confirm the fields you selected.

10. Press F3 (Exit) to save your query program.

11. Press the Enter key to run the query program.

The results of the query are shown on your display exactly as they appear in this example, and you can print these results as you would any physical file.

ON	PTF	LIC	PGM	RELEASE	OBJECT	STATUS
	ORDER				LIBRARY	
000002	N	SY00235	XXXXSS1	020100		Save file only
000003	N	SY00236	XXXXSS1	020100		Save file only
000004	N	EDL4000	XXXXSS1	020100	QCBL	Superseded
000005	N	EDL0022	XXXXSS1	020100	QLBL	Not applied
000006	N	EDL0021	XXXXSS1	020100	QLBL	Damaged

Figure 19. IBM Query Output

Testing PTFs before distribution

You may want to place newly received PTFs on hold so they can be tested to verify that they correct the reported problem. This makes managing your remote systems easier.

To hold a PTF save file so it cannot be distributed to a service requester when a PTF order is received, type HLDPTF at the command line.

To have PTFs placed on hold automatically when they are received over the service link or by using the Copy PTF to Save File (CPYPTFSAVF) command, set the hold PTFs received (HLDPTF) service provider attribute to *YES.

How the Hold Program Temporary Fix (HLDPTF) command functions

The Hold Program Temporary Fix (HLDPTF) command places PTF save files for supported licensed programs on hold. Use this command to hold PTFs that you may not want to distribute right away.

This command is valid only for supported products under one of the following conditions:

- If the Copy PTF to Save File CPYPTFSAVF command is used to create the PTF save file or if the PTF has been logged using the QPZLOGFX API.
- If the PTF is received by way of IBM Service (*IBMSRV) or a service provider.
- If the PTF is created using the Create Program Temporary Fix (CRTPTF) command.

Releasing previously held PTFs for distribution

After you have tested the PTF to make sure that it does fix the problem, be sure to release any PTF save files that are temporarily being held.

To release any PTF save files, type RLSPTF at the command line. This releases any PTF save files that were previously placed on hold.

How the Release Program Temporary Fix (RLSPTF) command functions

The Release Program Temporary Fix (RLSPTF) command releases PTF save files for supported licensed programs that were previously placed on hold. When a PTF save file is released and the send PTFs automatically (SNDPTFAUTO) service provider attribute is *YES, PTFs are automatically distributed to a service requester when a PTF order is received.

This command is valid only for supported products under one of the following conditions:

- If the Copy PTF to Save File (CPYPTFSAVF) command is used to create the PTF save file or if the PTF has been logged using the QPZLOGFX API.
- If the PTF is received by way of *IBMSRV or a service provider.
- If the PTF is created using the Create Program Temporary Fix (CRTPTF) command.

Deleting save files, cover letters, and on-order status

When you install a new release of the OS/400 licensed program on your system, unnecessary PTF save files may be on the system. If all the systems for which you provide service and support are also installing the new release, there will be save files and cover letters for PTFs from the previous release that you will want to delete from the system.

If you do not need to support systems running the previous release of the operating system, delete all the PTFs and associated PTF information. Use the Delete PTF (DLTPTF) command to delete all the PTFs for products you no longer support.

How the Delete Program Temporary Fix (DLTPTF) command functions

The Delete Program Temporary Fix (DLTPTF) command deletes PTF cover letters, save files, and on-order status. The DLTPTF command deletes PTF save files and removes PTF information from the PTF database. If a PTF is available for distribution and also loaded, the records indicating it is available are deleted. Records for PTFs that have been loaded and applied are kept until the product or product option is upgraded to the next release.

The Delete Program Temporary Fix (DLTPTF) command can be used to delete a list of PTFs, or delete all PTFs for a specific product and release.

To delete all PTFs for 1X12345 for V1R1M0 enter the following:

```
DLTPTF PTF(*ALL) LICPGM(1X12345) RLS(V1R1M0)
```

Cleaning up unnecessary PTF objects

When you install a new release of the Operating System/400 program on your system, unnecessary PTF objects may be left on the system. You can clean up these objects by entering the following:

```
CALL QPZDLOBJ
```

The physical files associated with the PTF database can be reorganized to reclaim storage automatically if you are using the Operational Assistant cleanup functions. If you are not currently using the Operational Assistant cleanup functions, see the procedure for setting up automatic cleanup under “Delete Problem (DLTPRB) command” on page 162.

Distributing PTFs

At times you may want to distribute PTFs to your service requesters without having the PTFs associated with a problem record. For example, you may want to have all your service requesters at the same maintenance level or you may want to prevent a known problem from occurring on your other service requesters.

One way to accomplish the distribution is to provide the operator at the service requester with the PTF numbers. The operator can then use the Send PTF Order (SNDPTFORD) command to order the PTFs. This causes a problem record to be created in the problem log and allows the PTFs to be automatically distributed as they would be in response to a service request.

When you want to distribute PTFs to a service requester without using a PTF order or service request, you can choose from the following procedures.

Sending PTFs to another site

You can service systems remotely by distributing PTFs and cover letters. The PTF to be distributed could have been created in any of the following ways:

- The PTF save file is created using the Copy PTF to Save File (CPYPTFSAVE) command or the QPZCPYSV API.
- The PTF is received by way of *IBMSRV or a service provider.
- The PTF is created using the Create Program Temporary Fix (CRTPTF) command.

To service your systems remotely, use the Send Program Temporary Fix (SNDPTF) to:

- Send a PTF to one or more service requesters
- Send a list of PTFs to one or more service requesters
- Apply PTFs on one or more remote systems
- Distribute any available cover letters
- Create customized PTF packages

The following is the display for the Send Program Temporary Fix command.

```

Send Program Temporary Fix (SNDPTF)

Type choices, press Enter.

PTF description:
PTF identifier . . . . . _____ Character value
Product . . . . . *SPT F4 for list
Release level . . . . . *ONLY *ONLY, VxRxMx
      + for more values
Destination service requester:
Network identifier . . . . . *SELECT Name, *ALL, *SELECT, *NONE ...
Control point . . . . . _____ Name
      + for more values
Maximum distribution . . . . . *NOMAX 1-5000, *SRVRQS, *SRVPVDA...
Send requisites . . . . . *YES *YES, *NO
Check service requesters . . . . . *YES *YES, *NO
PTF parts . . . . . *ALL *ALL, *PTF, *CVRLTR
Extent of change . . . . . *NONE *NONE, *TEMP, *PERM
Delayed PTF apply . . . . . *NO *NO, *YES,

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

```

To create customized PTF packages including PTF requisites, specify the *YES value for the send requisites (SNDRQT) parameter and *YES for the check (CHECK) parameter of the SNDPTF command. The system checks each service requester that will be receiving PTFs to determine which requisite PTFs are necessary to apply with the PTFs you want to distribute. Each service requester destination then receives only the PTFs that are necessary for that specific system.

When using the SNDPTF command and the service requester has Managed System Services installed, you can request that the PTFs be applied on the service requester. To choose how the PTFs are applied to the system, use the delay apply (DLYAPY) parameter. This immediately applies PTFs at the time the activity runs on the system or later during the next IPL.

When requesting that the PTFs be applied, you can also request an IPL of the service requester's system. You can schedule these operations to occur at a later date. You can manage your end of active jobs by specifying the power down option (PWRDWNOPT) and power down delay (PWRDWNDLY) parameters. A change request is automatically submitted that can be viewed to determine the status of the command. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.

The SNDPTF command uses the value you specify for the distribution queue on the Work with Service Requesters (WRKSRVRQS) command. *SVDS (SystemView distribution services) distribution queues are used for distributions between systems that use the Managed System Services licensed program. *SNADS (SNA distribution services) distribution queues are used within a SNADS network. You can override certain functions, and the queues can be held, released, and sent.

Language information

In PTF Packages the following information about national languages applies only to the SNDPTF, the RTVPTF, and the SNDPTFORD commands.

Sometimes when you specify or ask for a specific cover letter, you may receive a cover letter with a different language. For example, when you specify a cover letter of 2962 a search is started.

1. If the 2962 language is not found, then 2950 is sent or retrieved
2. If the 2950 language is not found, then 2924 is sent or retrieved
3. If the 2924 language is not found, then a cover letter is not sent or retrieved.

If you specify another language, a similar search is started.

1. If the requested language is not found, then language 2924 is sent or retrieved.
2. If the 2924 language is not found, then language 2950 is sent or retrieved.
3. If the 2950 language is not found, then no cover letter is sent or retrieved.

Working with PTFs remotely

You can work with remote PTFs using the commands provided with System Manager. Some of the commands provide parameter values to allow you to schedule these operations during off hours so as to have minimal effect on your daily system operation. The following items apply when you are working with remote PTFs:

- A change request is automatically submitted that can be viewed to determine the status of the command. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.
- A single service requester or a list of service requesters can be specified (exception on the Retrieve PTF command). Use the node list name parameter when you want to specify a node list object.
- It is required that the Managed System Services program be installed on the service requester to use remote PTF commands (for example, the Apply Remote PTF, Remove Remote PTF, Retrieve PTF, and Delete Remote PTF commands).
- If a node list object is used, you can have only entries with an *SNA address type.

Applying a remote PTF

The Apply Remote Program Temporary Fix (APYRMTPTF) command allows a service provider to remotely apply PTFs on the service requester's system. When you use this command, you need to specify the remote network identifier of the service requester as well as the date and time you want the PTF applied. The Apply Remote PTF (APYRMTPTF) can be scheduled to occur at a later date and time. To choose how the PTFs are applied to the system, use the delay apply (DLYAPY) parameter. This immediately applies PTFs at the time the activity runs on the system or later during the next IPL.

You can also request an IPL of the service requester's system. You can schedule the IPL during off hours so you do not interrupt daily operations. You can manage your end of active jobs by specifying the power down option (PWRDWNOPT) and power down delay (PWRDWNDLY) parameters.

Note: The system does not check distribution requisites. You must ensure that they are applied when you apply PTFs.

If a value is specified for the NODL parameter, the node list can contain only entries that have a value of *SNA for the address type.

Removing remote PTFs

The Remove Remote Program Temporary Fix (RMVRMTPTF) command allows a service provider to remotely remove PTFs from a service requester's system.

When using the Remove Remote PTF (RMVRMTPTF) command, you can request an IPL of the service requester's system. The remove PTF operation and the IPL can be scheduled to occur at a later date and time. To choose how the PTFs are remotely removed from the system, use the delay remove (DLYRMV) parameter. This immediately applies PTFs at the time the activity runs on the system or later during the next IPL. You can also request an IPL of the service requester's system. You can schedule the IPL during off hours so you do not interrupt daily operations. When you do request an IPL, you can manage your end of active jobs. To do this specify the power down option (PWRDWNOPT) and power down delay (PWRDWNDLY) parameters.

If a value is specified for the NODL parameter, the node list can contain only entries that have a value of *SNA for the address type.

Deleting remote PTFs

The Delete Remote Program Temporary Fix (DLTRMTPTF) command allows a service provider to remotely delete PTFs from a service requester's system.

If a value is specified for the NODL parameter, the node list can contain only entries that have a value of *SNA for the address type.

Retrieving PTFs

When you use the Retrieve Program Temporary Fix (RTVPTF) command, the following items apply:

- Only one service requester is allowed. A node list is not allowed.
- PTFs are retrieved from the service requester and received onto the service provider.

Creating PTF packages

The Create PTF Package command (CRTPTFPKG) provides an easier way of creating PTF packages than the CPYPTF command. Rather than specifying individual PTFs, this command operates against all PTF save files for all supported products and accepts output file PTF lists for select and omit control. The input files for this command must have the same format as the output file created by the Display PTF (DSPPTF) command.

An output list of PTFs is created, and optionally, the PTFs are copied to a magnetic tape. When creating large packages, it is recommended to first specify *NONE for the device (DEV) parameter. This allows the package output file to be reviewed before saving to tape.

PTFs are not included in the package for the following reasons:

- The PTF is for a product option that is not supported
- The PTF save file is in a held status
- A save file does not exist for the PTF
- The PTF is listed in the omit list
- The PTF has been superseded and *YES is specified for the replace superseded PTFs (SUPERSEDE) parameter.

Examples for creating PTF packages

The following example creates a list of all PTFs, which have not been superseded, for all supported products. The list is written to member OUTMBR1. The resulting PTFs are copied to tape. A cover letter for NLV 2924 is also included on the tape for each PTF. Any PTF save files that have a status of held are excluded from the PTF list.

```
CRTPTFPKG OUTFILE(LIB1/OUTFILE) +
          OUTMBR(OUTMBR1) +
          SELECT(*SUPPTD) +
          SUPERSEDE(*YES) +
          PREREQ(*ALL) +
          DEV(TAP01) +
          COVER(2924)
```

After a period of time, you can distribute a new fix tape that only contains PTFs created since the previous one was distributed. The outfile created in the previous example is specified on the OMIT parameter. This causes all PTFs that were previously distributed to be excluded from this distribution. In this example, the PTFs are not copied to tape. The only output is the list of PTFs in the output file.

```
CRTPTFPKG OUTFILE(LIB1/OUTFILE) +
          OUTMBR(OUTMBR2) +
          SELECT(*SUPPTD) +
          OMIT(LIB1/OUTFILE OUTMBR1) +
          SUPERSEDE(*YES) +
          PREREQ(*ALL) +
          DEV(*NONE)
```

In this example, all PTFs for product 1EXPROD are excluded from the list that was previously created. An output file that lists all PTFs for 1EXPROD is created using DSPPTF. This list is specified on the omit parameter of CRTPTFPKG, so those PTFs are excluded.

```
DSPPTF LICPGM(1EXPROD) +
       OUTPUT(*OUTFILE) +
       OUTFILE(LIB1/OUTFILE) +
       OUTMBR(OMITPTFS)
```

```
CRTPTFPKG OUTFILE(LIB1/OUTFILE) +
          OUTMBR(OUTMBR3) +
          SELECT(LIB1/OUTFILE OUTMBR2) +
          OMIT(LIB1/OUTFILE OMITPTFS) +
          DEV(TAP01) +
          COVER(2924)
```

Cumulative PTF packages

The following example describes how cumulative PTF packages can be created, distributed to remote systems, and loaded and applied on those systems.

1. Create an output file with a list of all PTFs already on the remote systems. If the central site system has a different PTF level than the remote systems, then run this command on a remote system and send it to the central site system.

```
DSPPTF OUTPUT(*OUTFILE)
       OUTFILE(QGPL/PTSTHERE MAY94)
```

2. Ensure that the PTFs to be included in the package are released.
3. Verify that the products for which PTFs are to be sent are supported. Only those products that are supported using the Work with Supported Products (WRKSPTPRD) command are sent to the remote systems. When you type wrksptprd on the command line and press the Enter key twice, the following display is shown.

```

Work with Supported Products
System: XXXXXXXX
Release: *CURRENT

Type options, press Enter.
4=Remove support 5=Display 6=Print
8=Work with supported language features

Product
Opt Product Option Description
57XXXXX *BASE Operating System/400
INFOAS4 *BASE Information APARs
INFODSL *BASE Information APARs
5769XXX *BASE Advanced Function Printing Utilities/400
57XXXXX *BASE Advanced DBCS Printer Support/400
57XXXXX *BASE Backup Recovery and Media Services/400
More...

Command
===>
F3=Exit F5=Refresh F6=Print list F11=Display releases F12=Cancel
F14=Support additional products F17=Position to

```

4. Receive the cumulative PTF package from IBM. Copy the PTFs from the distribution media onto the system using the CPYPTFSAVF command. You can load and apply the PTFs at the central site system if you want.

```
CPYPTFSAVF FROMDEV(TAP01)
```

5. Use the Create Program Temporary Fix Package (CRTPTFPKG) command to create a list of the PTFs that need to be sent to the remote sites. The list consists of all PTFs on the central site. This including the new cumulative PTF package. This list includes prerequisites, corequisites, and supersede PTFs of those that are supplied to the command. This list of PTFs will be used in the next step to do the Copy Program Temporary Fix (CPYPTF) command.

```

CRTPTFPKG OUTFILE(LIB/PTFSTOSEND)
OUTMBR(MAY94)
SELECT(*SUPPTD)
OMIT(QGPL/PTSTHERE MAY94)
SUPERSEDE(*YES)
PREREQ(*ALL)
DEV(*NONE)

```

6. The Copy Program Temporary Fix (CPYPTF) command is used to create a save file containing many PTFs. It takes as input a list of PTFs for a single product. Use the list from the previous step to create a save file for each product that is supported. The CPYPTF command uses a list with a maximum of 50 PTFs. You will need to create multiple save files if there are many new PTFs for a product.

```

CPYPTF LICPGM(PROD1)
FROMDEV(*SERVICE)
TODEV(*SAVF)
SELECT(SI12345 SI9080 SI57657 SI78964...) <== example numbers
TOSAVF(LIB/PTFPKG1)

```

```

CPYPTF LICPGM(PROD2)
FROMDEV(*SERVICE)
TODEV(*SAVF)
SELECT(SI12455 SI3670 SI52677 SI78226...) <== example numbers
TOSAVF(LIB/PTFPKG2)

```

```

.
.
.

```

Notes:

- a. You could write a CL program to issue all of the load and apply commands on the remote system.
 - b. Some PTFs on the cumulative package are to make changes to text information (for example, a message). When this occurs there is a unique PTF for each of the 29 languages supported by iSeries server. Only one of these PTFs should be copied using CPYPTF. The output file contains a load ID field that indicates it is associated with a particular language (for example, 2924 English) and can be used to select only those PTFs for the specified language.
7. Send the save files containing the PTFs to the remote site systems.
 8. Issue a Load PTF (LODPTF) command for each save file.

```
LODPTF LICPGM(PROD2)
      DEV(*SAVF)
      SAVF(LIB/PTFPKG1)
```

```
LODPTF LICPGM(PROD2)
      DEV(*SAVF)
      SAVF(LIB/PTFPKG2)
```

```

      .
      .
      .
```

9. Issue an Apply PTF (APYPTF) command for *ALL or for each product.

```
APYPTF LICPGM(PROD1)
      DELAYED(*IMMDLY)
```

```
APYPTF LICPGM(PROD2)
      DELAYED(*YES)
```

```

      .
      .
```

Note: Products like System Manager can be used to distribute the save files and to run the load and apply commands.

Chapter 7. Servicing your products using System Manager

Because service can vary from one product to the next, the product owner must tell the product user how to obtain service for the product.

This chapter is divided into two sections. The first section provides information that the product owner needs to know. The second section contains information that the product user needs to know. Product owners must consider the information in the second section of this chapter when developing service procedures for their product. It is important that product developers understand what PTF tasks and functions are familiar to their users.

The functions that are described in this manual apply not only to IBM licensed programs, but also to applications that are packaged using the Create PTF (CRTPTF) command.

Service information for the product owner

This section provides information for the owner of the product. This section describes the process for servicing a packaged product that is based on the program temporary fix (PTF) functions that are provided by the Operating System/400 program. This chapter provides the following information:

- Getting your system ready to create PTFs for your own products
- Using the Create PTF (CRTPTF) command
- Establishing libraries for your PTFs
- Adding and replacing objects in your products
- Establishing prerequisite, corequisite, and superseded PTF relationships
- Preparing cover letters
- Searching the problem log for symptom strings
- Changing documents in folders
- Developing exit programs to reduce manual operations
- Developing PTFs for directories in the Integrated File System

Getting your system ready to create PTFs

Before creating any PTFs, resolve all OS/400 release issues. The following steps contain instructions regarding creating PTFs to be distributed to a previous release:

1. Before you create a PTF that is distributed to a previous release of the operating system, ensure that all program objects for the PTF are created properly. Specify the system on the target release TGTRLS(*PRV) parameter for the Create PTF (CRTPTF) command.
2. When you create the PTF specify the earliest release of the operating system that this PTF will be applied. Specify the target release (TGTRLS) parameter of the Create PTF (CRTPTF) command.

You can only create PTFs for the operating system releases that are supported by the save commands (for example, SAVOBJ, SAVLIB, and others).

Verify that your product or PTF has been saved for installation on a previous release. Use one of the following commands to check the release level field:

- Display Save File (DSPSAVF) command
- Display Tape (DSPTAP) command

Note: If you have copied PTFs to tape, then use the Display Tape (DSPTAP) command. The Display Tape (DSPTAP) command determines the release for which the PTF is targeted.

Creating a program temporary fix

You only can create Program Temporary Fixes for products that are installed. If you have not installed your product since its last packaging, use the Restore Licensed Program (RSTLICPGM) command to install your product. You must do this before creating any PTFs. The Create PTF (CRTPTF) command requires the following product information be specified for each PTF:

- Product ID (LICPGM parameter)
- Option ID (OPTION parameter)
- Load ID (LODID parameter)
- Release level (RLS parameter)

All the objects for a PTF must share the same product, option, load ID, and library. You cannot repair objects for two different options or loads within the same PTF. If the PTF affects human-readable information that has been translated, then a different PTF must be created for each national language version (NLV). The national language version is specified on the LODID parameter.

How CRTPTF command functions on your system

The Create PTF (CRTPTF) command produces a PTF save file (and optional cover letters) in the general purpose library (QGPL). The save file contains a PTF control object and any number of fix objects. The save file name is the PTF identifier that is preceded by a "Q". If a file with the same name already exists in QGPL, then the system creates a unique name. A unique name contains a "Q" and a timestamp that are assigned by the system. After creating a PTF, you can use the display option of the Work with PTF command (WRKPTF) to view the PTF attributes.

PTFs must be created by a profile that is known to exist on the user's system. This user profile must have adequate authority to perform all the PTF functions because you must test the PTF thoroughly before distribution.

You can create PTFs for a previous release of OS/400. You can send, load, and apply a PTF to a system with a previous release installed.

Assigning an identification name to your program temporary fixes

To track your PTFs, assign an identification name or PTF identifier to any PTFs that are created using System Manager. A PTF identifier contains seven characters; the first character must be numeric. For example: 1A00015. IBM PTFs start with a character. You must also enter the product and release level. For each product and release level, only once can you use the same PTF identifier. To effectively use the Display PTF (DSPPTF) command, assign your PTF identifiers in ascending numeric sequence. This displays the most recent PTFs first.

You can use the Work with PTF (WRKPTF) command to assign (or change) the PTF range for each release level of your product. Once you have defined a range, all PTF IDs for the product must be within the range.

Establishing libraries for your program temporary fixes

Specify two library names for the PTFs that you are creating when using the object library (OBJLIB) parameter within the Create PTF (CRTPTF) command. The first library, the **PTF development library**, is where the create PTF function finds objects for inclusion in the PTF. This can be any library you create. The possible values for the development library are *CURLIB or a development library name. The other library, the **PTF primary library**, is usually where PTF objects are placed when the PTF is installed. The possible values for the primary library are *PRINCIPAL and a primary library name. The default for the primary library name is *PRINCIPAL. Change this value if the option being repaired exists in more than one library, and the objects reside in a different library than the principle library.

If necessary, the PTF apply operation automatically overrides the PTF library that is specified when the PTF was created. Two cases where this might occur include:

- The product load has been installed into a secondary language library.
- Dynamic library naming was used when the product load was installed.

Replacing and adding objects to your product using program temporary fixes

You can add new product function or revise existing function by using PTFs. To create your own PTFs, use the Create PTF (CRTPTF) command. You can replace or add specific objects to your product by specifying the PTF object (PTFOBJ) parameter. If the object that is included in the PTF already exists in the library where the PTF is being installed, then the PTF object replaces the existing object.

PTFs cannot contain certain object types. These object types cannot be saved, restored, renamed, duplicated, moved, or deleted as required by the PTF process.

A PTF object type that has been rejected by the Create PTF (CRTPTF) command can usually be fixed by using a PTF exit program and a temporary object. The exit program can create the object when the PTF is applied. The exit program should be a temporary PTF object if it is not needed after the PTF has been applied.

A temporary PTF object is deleted by the apply PTF permanent operation. To define a temporary object, give it a name that begins with the characters "QPZ1" (for example, QPZ1xxxx). For a given release level of your product, a temporary object name should only be used once. If the same temporary object is specified for two different PTFs, those PTFs take the place of each other.

The PTF includes the object owner and authorizations. Ensure that the owner and authorized user profiles exist on all systems where the PTF is to be applied. Create the PTF using the profile that is to own the objects in the PTF.

Establishing dependencies on other PTFs

Three types of relationships can exist between PTFs. A **prerequisite relationship** is when one PTF requires that another PTF also be applied. A **corequisite relationship** occurs when the PTF that you are creating requires two or more PTFs that are applied together as a set. Sometimes one PTF can replace another PTF. This condition describes a **superseded relationship**.

Establishing prerequisite PTF relationships

Establish a prerequisite relationship when the PTF that you are creating requires another PTF to be previously applied on the system. Use the Create PTF (CRTPTF) command to specify prerequisite PTFs parameter (PREREQ). Prerequisite PTFs

must exist within the same product. PTFs in other products cannot be specified as prerequisites. A prerequisite PTF must meet the following conditions:

- Exist on the system
- Belong to the same option or base option of the same product

If either of these conditions is not met, the Create PTF (CRTPTF) command fails.

When a PTF which is created with a prerequisite PTF specified, that PTF is a dependent PTF. A dependent PTF can be applied on the same command invocation as its prerequisite, or it can be applied later. If they are applied on the same command invocation the system applies the prerequisite first, within that invocation.

Establishing corequisite PTF relationships

Establish a corequisite relationship when you are creating two or more PTFs that are required to be applied together as a group. PTFs that have a corequisite relationship ensure that the PTFs are applied along with their corequisite PTFs. However, the order in which you apply the corequisite PTFs does not matter as long as you apply them at the same time. Because the system may apply corequisite PTFs in any order, it is important that you do not create corequisite PTFs that must be applied in any sequential order. To create a PTF, type CRTPTF at the command line, and specify the parameters as needed. List the related PTFs that must accompany the PTFs that you are creating in the COREQ parameter field. A corequisite PTF belongs to the same product, release, option, and load. Corequisite PTFs must identify each other as corequisites. In advance, you must decide to determine a PTF identifier for all the corequisite PTFs.

Establishing superseded PTF relationships

Establish a superseded relationship when the PTF that you are creating replaces or supersedes another PTF. The CRTPTF command automatically identifies superseded PTFs. The objects that you specify on the PTFOBJ parameter are used to create a list of superseded PTFs. The save file containing superseded PTFs must exist on the system or the Create PTF (CRTPTF) command will fail. If a PTF contains at least one matching object, when the PTF database is searched, it is added to the superseded list. Any other objects in the superseded PTF are added to the object list and are copied into the PTF save file being created. This is done so that only the most recent PTF in a supersede chain has to be applied. When this PTF is applied, all prior fixes to the object and all related fixes are applied at the same time. The exit programs are also brought forward and added to the end of the exit program list. They will be the last exit programs to be called when you apply or remove a PTF.

Preparing PTF cover letters

A cover letter can be created for each of the national language versions that IBM supports. Each cover letter is stored as a member of the QGPL/QAPZCOVER file. A member that contains source information for each PTF cover letter must be specified on the CRTPTF command (parameter COVER). The cover letter input file can be a source file with maximum record length 92 or a physical file with record length 80.

The following information is added to your cover letter input by the Create PTF (CRTPTF) command:

- PTF identifier
- Product
- Release

- Option
- Load ID
- PTF object library
- Corequisite PTFs
- Prerequisite PTFs
- Superseded PTFs

You can also create PTF cover letters by specifying just the cover letter parameter on the CRTPTF command.

Headings for the various sections of the cover letter originate from message files on your system. To create a PTF cover letter in a particular NLV, you need the NLV installed for the OS/400 program.

Notice which character set displays your PTF cover letter. It is generally best to use a file with the same character set for input that is used to display the cover letter. Another option is to use only invariant characters.

For instance, if you are creating a French cover letter for a French customer, you may want to create the source file with a coded character set ID of 297. When your customers display the cover letter on their system, it uses the coded character set ID of 297, and loses no data.

Searching the problem log

You can automatically search the problem log for any symptom strings that are associated with the PTF ID. Any symptom strings that are found are included in the PTF. If you want to use the System Manager function that automatically handles service requests, you must assign PTF identifiers to problems before you use the Create PTF (CRTPTF) command.

Use the following sequence to create a problem record for a known problem:

1. If the Analyze Problem (ANZPRB) command reports the problem, use the ANZPRB command to create a problem record.
2. To create a problem record, call the API directly or write a program to call the API, if the problem is reported by the First Failure Data Capture API.

Use the following sequence to assign a PTF identifier to a problem record before you create the PTF:

1. Use the Work with Problems (WRKPRB) command to display a list of problems
2. Select option 8 to work with the problem
3. Select option 5 to answer the problem
4. Select option 2 to work with the PTF information
5. Select option 1 to add a PTF to the problem
6. Enter the requested information

Note: You may first need to add a range by using the Work with PTF (WRKPTF) command or by pressing function key 19 (F19).

If you specify *GEN, the next unused PTF identifier from the range is used for the product and release level that you specified.

When you are ready to create a PTF that is assigned to one or more problem records, use the Work with PTF (WRKPTF) command (F18). This function lists all assigned PTF identifiers for a product and release level. You can select any PTF ID from this list so that the Create PTF (CRTPTF) command prompt displays all the known information.

When you want to install a PTF, use the Work with PTF (WRKPTF) command (F20).

Using program temporary fixes to change documents inside folders

If your product definition contains one or more folders, you may create PTFs to change the documents in the folders. Use the PTFDOC parameter to specify the document and folder names that you want to change with the PTF. The create PTF function copies the document from a subfolder with the name that is specified and “/QP” added. If a PTF is being created for a product folder called PRODUCT, then develop the fix in a subfolder by the name of PRODUCT/QP. Note that the object, however, is installed during the apply PTF operation into the product folder PRODUCT. The QP subfolder allows you to develop a PTF without changing the product. The product only changes when you apply the PTF to test it.

Using program temporary fixes to change stream files inside directories

If your product was defined with one or more directories, you can create PTFs to change the stream files in the directories within the Integrated File System.

Establishing directories for development

When you are using the Create PTF (CRTPTF) command to create PTFs that service products which have directories, you must specify two directories on the directory object (DIROBJ) parameter. First, you must specify the product directory for your PTF. This is the directory where the PTFs are placed or installed into after you are finished developing your PTF. This is the directory that is defined in the product. Second, you must specify the name of the development directory. This is the directory where you develop changes by creating the PTF function. Third, you must specify the objects that are included in the development directory which belong to the PTF that you are creating.

Determining naming conventions for PTFs inside directories

To correctly create your PTF using the Create PTF (CRTPTF) command, you must carefully determine the path name in the DIROBJ parameter field. There are some naming restrictions. You can specify up to a total of 30 product home directories and product directories. You are restricted to specifying directory names that contain the following:

- Precede your path name with a forward slash (/).
- Name your path names using upper case (A-Z) and lower case (a-b) alphabets and numeric values (0-9).
- You can use the following symbols in the directory names:
 - Comma (,)
 - Period (.)
 - Colon (:)
 - Semi Colon (;)
 - Ampersand (&)
 - Plus sign (+)

- forward slash (/)
- Percent sign (%)
- Equal sign (=)
- Greater than sign (>)
- Less than sign (<)
- Right Parenthesis)
- Left Parenthesis (
- Minus sign (-)
- Underscore (_)

Following are some restrictions for directories:

- Do not specify /QSYS.LIB and /QDLS directories for the directory object (DIROBJ) parameter.
- Do not end path name with a forward slash (/).
- Do not use a blank in the directory path name.
- Do not use the following character combinations of ../ and ./ for any directory path names.

Developing PTF exit programs to eliminate manual operations

When creating a new PTF you can specify one or more PTF exit programs. Use a PTF exit program when a PTF is temporarily applied, permanently applied, temporarily removed, or permanently removed. To call the exit program, you must specify the run option field of the EXITPGM parameter. Exit programs eliminate the need for you to manually carry out special instructions to install the PTF.

Another way to avoid unwanted supersedes is to ship the exit program with a different temporary object name in each PTF. When the exit program is being shipped with the PTF, specify the value *PTF for the Type field of the EXITPGM parameter. Shipping the same exit program in two PTFs for the same product causes one PTF to supersede the other. Avoid this by including the PTF exit program when you initially package the product. If a PTF exit program already exists in the product, then specify the *OBJLIST value on the EXITPGM parameter for the Type field.

Apply PTF exit programs are called after the objects in the PTF have been renamed from the QPZRxxxxxx name to the original object name. Remove PTF exit programs are called after the objects in the PTF have been renamed from the original object name to the QPZRxxxxxx name.

Pre-apply exit programs are called before the objects are renamed and again after the objects are renamed. Pre-remove exit programs are called before the objects are renamed and after the objects are renamed.

If a PTF is applied permanently from a loaded state, the exit program runs twice. It is called once for the temporary apply and once for the permanent apply. If a temporary applied PTF is permanently removed, the exit program is called twice. It is called once for the temporary remove and once for the permanent remove.

Displaying details about a program temporary fix

The Apply Program Temporary Fix (APYPTF) command applies to fixes of a specified licensed program. To display the program temporary fixes for a specified licensed program, use the DSPPTF command. The following is an example of this command.

```
Display Program Temporary Fix (DSPPTF)          XXXXXXXX0
Type choices, press Enter.
Product . . . . . *ALL F4 for list
PTF numbers to select . . . . . *ALL Character value, *ALL...
Release . . . . . *ALL *ALL, VxRxMx
Cover letter only . . . . . *NO *NO, *YES
Output . . . . . * * , *PRINT, *OUTFILE

Bottom

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

Applying a program temporary fix

When a PTF is applied, it completely replaces the affected objects in the licensed program. PTFs can be applied temporarily or permanently.

If the PTF is applied temporarily, the system saves the replaced objects. You can restore the objects later using the Remove Program Temporary Fix (RMVPTF) command. If the PTF is applied permanently, the system deletes the replaced objects.

You can use the Apply Program Temporary Fix (APYPTF) command in two ways:

- To apply immediate PTFs when the command is running.
- To request PTFs to be applied during the next unattended initial program load (IPL).

Testing PTFs

Before you distribute or release your PTF, make sure that you perform the following tests:

1. Read the cover letter and display the PTF using the Work with PTF (WRKPTF) command.
2. Using the Work with PTF (WRKPTF) command, you can:
 - a. Install (check exit program for temporary apply results) PTFs
 - b. Temporarily remove (check exit program results) PTFs
 - c. Permanently remove (check exit program results) PTFs
 - d. Verify that exit program handles any error conditions
 - e. Permanently apply PTFs
3. Check object owner and authorized user profiles using the Display Object Authorization (DSPOBJAUT) command

4. Test the fix
5. Apply the object permanently by using the Apply PTF (APYPTF) command (check exit program results)

If your PTF is adding a new permanent object to the product after permanently applying the PTF, check to see that the object still exists. If your PTF uses a temporary object, you should verify that the object no longer exists after the PTF has been permanently applied.

Rebuilding PTFs

There is no function to change a PTF. If you find a problem when you are testing:

1. Permanently remove the PTF if it has already been applied or loaded.
2. Use the Delete PTF (DLTPTF) command to delete the PTF.
3. Then, create the PTF again.

If you find a problem with the PTF after it has been permanently applied and the PTF has not been distributed, delete the PTF save file. To delete the PTF save file, use the Work with PTF (WRKPTF) command. To clean up any records of the PTF, restore the product option by using the Restore Licensed Program (RSTLICPGM) command. This allows the PTF to be created again using the same PTF ID.

If you find a problem after you distribute the PTF, create a superseding PTF. Do not delete the bad PTF until you have created the superseding PTF. The Create PTF (CRTPTF) command needs to extract information from the save file of the superseded PTF.

Backup and recovery of PTF information

Backup and recovery procedures for service providers should include PTFs. Backup and recovery procedures for any system on which PTFs are created or kept for distribution should include PTFs, regardless of whether the system is part of a network.

The Copy PTF (CPYPTF) command can be used to save the PTFs to tape. To restore the PTFs back to the system, use the Copy PTF to Save File (CPYPTFSAVF) command.

The Send PTF (SNDPTF) command can be used to back up PTFs to another system in the network.

If the system that you use to create your PTFs has to be recovered, you will need a backup that includes all of your PTFs. This is necessary so that the Create PTF (CRTPTF) command can properly maintain the PTF supersede lists.

Distributing PTFs

When the PTF is ready to distribute, use the release option of the Work with PTF (WRKPTF) command to release the PTF save file. PTF save files can be distributed in the following ways:

- Service provider broadcasts the PTF to the service requester by using the Send PTF (SNDPTF) command.
- The Create PTF Package (CRTPTFPKG) command copies PTFs to tape for distribution by mail.
- Service requester uses the Send PTF Order (SNDPTFORD) command and receives the PTF.

- Service requester reports a problem and receives a PTF through the Analyze Problem (ANZPRB) command or the Work with Problem (WRKPRB) command.
- The APYRMTPTF command allows you to apply a PTF on a remote system.
- The RMVRMTPTF command allows you to remove a PTF on a remote system.
- With Management Central. For more information about distributing PTFs with Management Central, see the Software and licensed program topic under the Systems Management category in the Information Center.

Product user information for PTF developers

This information is provided to product owners for two reasons:

- So product owners can tell users how to request service
- So product owners can service their own products

Analyzing problems

The Analyze Problem (ANZPRB) command helps determine the cause of a problem. Additionally, this command suggests a possible PTF needed to fix the problem. A series of displays is presented for entering information to identify the problem. The information is formatted into a symptom string. A record is created in the problem log and the symptom string is recorded there.

Do the following steps to create a problem log record with an associated symptom string:

1. Type ANZPRB and press the Enter key.
2. Type 1 (for this AS/400 system) and press the Enter key.
3. Type 1 (for job or program problem) and press the Enter key.
4. Type 1 (for incorrect output) and press the Enter key.
5. Type 2 (for incorrect output) and press the Enter key.
6. Type 2 (to create a problem record) and press the Enter key.
7. Type 1 (to select the product) and press the Enter key.
8. Type 1 (to select the product option) and press the Enter key.
9. Type in one or more fields on this display and press the Enter key.

What you enter here becomes the symptom string. The product owner provides instructions on what to enter. For example, the message line and from-program field may be the only pertinent information for a particular product.

10. Type in a brief description in your own words and press the Enter key.
11. Press F3 (to exit ANZPRB).

If the problem record is created on a service provider, the symptom string search occurs immediately. If a match is found for the product and option specified, the PTF is added to the problem log record. Use the Work with Problem (WRKPRB) command to view the problem record, symptom string, and any PTFs.

You can use the ANZPRB command to display the Analyze a Problem menu.


```

                                Analyze a Problem
                                System:  RCHAS990
Select one of the following:

Analyze a Problem
1. Job or program problem (application or system)
2. System performance problem
3. Hardware problem
4. Communications/LAN hardware problem

Describe a Problem
5. Problem occurred during IPL of this AS/400 System
6. Job or program problem (application or system)

Selection or command
=====
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel

```

Reporting problems

When the problem record is created on a service requester, the problem must be reported to the service provider. The service provider performs the symptom string search. If any PTFs found for the product contain the same symptom string, they can be sent back to the service requester.

When your application detects a problem, it can use the First Failure Data Capture (FFDC) function. The FFDC creates a problem record without a thread environment. It builds a symptom string, and collects data. The system uses the symptom string to search for existing PTFs. If the system cannot find a PTF, the data may help to quickly resolve the problem.

Installing PTFs

Refer to the Software and licensed program topic under the Systems Management category in the Information Center for general information about correcting potential problems or adding new function using PTFs. This topic explains installing, displaying, loading, applying, and removing PTFs. It also explains additional information about using cover letters. For a list of PTF commands that are available to all your product users, type GO PTF at the command line. Explain any unique information about installing a particular PTF in the PTF cover letter.

Remember that each user profile that applies PTFs must have the proper authority. To apply PTFs for your product, each product owner should specify what authorities are needed.

PTF apply usually replaces objects. This may adversely affect the operation of the product. The product owner should document any requirements and procedures for determining when it is safe to apply PTFs to their product while the product is being used.

Note: When you load PTFs from a tape, typing GO PTF at the command line does not support PTFs for products with multiple releases that are installed.

Saving and restoring PTFs

Use the Save Licensed Program (SAVLICPGM) command to save PTF objects and statuses. Use the Restore Licensed Program (RSTLICPGM) command to restore PTF objects and PTF statuses. Use the Save Library (SAVLIB) and Restore Library (RSTLIB) commands to save and restore PTF objects and PTF statuses.

Cleaning up PTF information

When a new release of your product is installed into the same library as the old release, the PTF information for the old release is deleted. The Delete Licensed Program (DLTLICPGM) command cleans up PTF information for any product or option that is deleted. Use the Delete PTF (DLTPTF) command to delete individual PTF save files and cover letters. PTFs that have been loaded can only be removed from the system by using the Remove PTF (RMVPTF) command. The record of permanently applied PTFs is only removed when the next release is installed or the product option is deleted.

Note: Always use the Delete PTF (DLTPTF) command to delete PTF save files.

Chapter 8. Working with the Problem Log

The problem log is a useful tool for coordinating and tracking all your problem management operations. The main purpose of the problem log is to hold problem, change, configuration, and user-defined information. From the problem log at the service provider, you can perform most of the operations necessary for problem management in a network environment. If you do not want to access the information interactively, you can use applications or APIs to access the database.

The problem log at a service provider contains problem records created as a result of:

- Received incoming alerts
- Received service requests and program temporary fix (PTF) orders
- Local system-detected problems
- Local user-detected problems (Analyze Problem (ANZPRB) command)

Alerts cause problem records to be created only when the log remote problems (LOGRMTPRB) service provider attribute is set to *ALERT or *ALL.

Note: Only an alert that already has a problem record existing on the service requester can cause a problem record to be created on the service provider. Service requests and PTF orders cause problem records to be created only when the log remote problems (LOGRMTPRB) service provider attribute is set to *SRVRQS or *ALL. See "Setting the service provider attributes" on page 18 for more information about service provider attributes.

When the system detects a problem, a problem record is created automatically. When a user detects a problem, a problem record is created in one of two ways:

1. Using the Analyze Problem (ANZPRB) command and selecting an option to create a problem record
2. Using the GO USERHELP command and selecting option 10 (Save information to help resolve a problem)

A problem record contains information about the type and origin of a problem, as well as the results of any problem analysis that has been performed. The following displays provide examples of the information that may be found in a problem record when a device error occurs.

Display Problem Details

System: HQSYS

Problem ID : 9517241463
Origin : CSTNET1.CHICAG01
Current status : READY
Problem : Software problem data for QTOACLS has been
Refer to help text for additional information.

Problem message ID : CPI93B0
Problem type : Machine detected
Problem category : *REPORT
Date and time detected : 06/21/95 12:04:37
System reference code : SRCB900FDC5

More...

Press Enter to continue.

F3=Exit F6=Display problem history F11=Display APAR library F12=Cancel
F16=Display spooled files F20=Display submitted change request

Display Problem Details

System: HQSYS

Problem ID : 9517241463
Origin : CSTNET1.CHICAG01
Current status : READY
Problem : Software problem data for QTOACLS has been
Refer to help text for additional information.

Reporting program ID : 5763SS1
Version : 0003
Release : 06
Modification level : 00
Program : QTOACLS
Service ID : 9001

More...

Press Enter to continue.

F3=Exit F6=Display problem history F11=Display APAR library F12=Cancel
F16=Display spooled files F20=Display submitted change requests

```

Display Problem Details
System:  HQSYS

Problem ID . . . . . : 9517241463
Origin . . . . . : CSTNET1.CHICAG01
Current status . . . . . : READY
Problem . . . . . : Software problem data for QTOACLS has been
Refer to help text for additional information.

Origin system . . . . . : 9406
Model . . . . . : 510
Feature . . . . . : 2144
Serial number . . . . . : 10-08BCD
Program ID . . . . . : 5769SS1
Version . . . . . : 0004
Release . . . . . : 02
Modification level . . . . . : 00
Level . . . . . : 00

More...

Press Enter to continue.

F3=Exit  F6=Display problem history  F11=Display APAR library  F12=Cancel
F16=Display spooled files  F20=Display submitted change requests

```

The function keys provided depend on the type of problem and the operations that have been performed. Using the function keys, you can look at the possible causes determined by problem analysis, the history of operations performed for the problem at the service provider, and the contact information (service overrides) for the system that reported the problem.

Analysis results shown on the display indicate the problem is recognized by the system and show specific information about the system with the problem, the area (device) where the problem occurred, and the symptom string that is used to search for PTFs. However, for a more complete listing of the information gathered by problem analysis, you can use the function key provided (F5=Display possible causes) to display the possible causes of the problem.

```

Display Problem Details
System:  HQSYS

Problem ID . . . . . : 9517241463
Origin . . . . . : CSTNET1.CHICAG01
Current status . . . . . : READY
Problem . . . . . : Software problem data for QTOACLS has been
Refer to help text for additional information.

Detecting program ID . . . . . : 5763SS1
  Version . . . . . :
  Release . . . . . :
  Modification level . . . . . :
  Program . . . . . : QTOACLS
Symptom string . . . . . : 5763 F/QTOACLS MSG003419Errno RC1

APAR library . . . . . : QSC7241463

More...

Press Enter to continue.

F3=Exit  F6=Display problem history  F11=Display APAR library  F12=Cancel
F16=Display spooled files  F20=Display submitted change requests

```

If the problem has been answered and a change to software or Licensed Internal Code is necessary, a function key is provided (F9=Display PTFs) to display information about the PTFs that will correct the problem.

```

Display Problem Details

Error log ID . . . . . : 80000475
Original reference code . . . . . : FDC5
  Description . . . . . : Error detected in OS/400 licens

Translate table . . . . . : QSYS/QB900FA00

Press Enter to continue.

More...

F3=Exit  F6=Display problem history  F11=Display APAR library  F12=Cancel
F16=Display spooled files  F20=Display submitted change requests

```

When spooled files are associated with a problem, pressing the function key to display spooled files shows a list of the names of the related spooled files. If the problem occurred on the local system, you can display the file contents. If the system where the problem occurred is not the local system, then these files are not available and you need to use display station pass-through or use another method to get these files from the remote system. If you try to display the files and they are on a remote service requester, you will receive a message indicating the files cannot be found.

```

Display Problem Details                                     System:  HQSYS

Problem ID . . . . . : 9517241463
Origin . . . . . : CSTNET1.CHICAG01
Current status . . . . . : READY
Problem . . . . . : Software problem data for QTOACLS has been
detected. Refer to help text for additional information.

Failing program ID . . . . . : 5763SS1
  Version . . . . . : 0003
  Release . . . . . : 06
  Modification level . . . . . : 00
  Program . . . . . : QTOACLS
  Failing Instruction . . . . . :
  Function . . . . . :
Alertable condition . . . . . : Yes
Number of occurrences . . . . . : 2

Press Enter to continue.

Bottom

F3=Exit  F6=Display problem history  F11=Display APAR library  F12=Cancel
F16=Display spooled files  F20=Display submitted change requests

```

The information in a problem record helps you determine the cause of the problem and allows you to make the decisions necessary to solve it.

The following sections discuss the commands necessary to display and work with the problem log. The options available to work with a particular problem are presented, followed by an explanation of the status changes that occur when working with a problem record. Finally, a brief description of problem tracking is presented, followed by an explanation of how to work with PTF information for a specific problem record.

Using the problem log commands

The following explanations of the problem log commands do not provide descriptions of the individual parameters.

Work with Problem (WRKPRB) command

To work with a problem or problems from the list of problems existing in your system's problem log, type the Work with Problem (WRKPRB) command on any command line.

When you run the WRKPRB command, the Work with Problems display is shown with a list of all the problems logged on the system or a specific group of problems you requested when you specified the parameter values. From the Work with Problems display you can:

- Change various values assigned to a problem record
- Delete a problem record
- Display or print the details about a problem
- Work with an individual problem (problem management tasks)
- Work with the alert associated with a problem
- Enter or edit your own notes about a problem
- Work with change request descriptions

You can work with change requests and change request descriptions. There are two events records. One is to confirm that the change request is submitted and the other is to confirm that the request has ended or has completed.

The following displays show the default values for the parameters of the WRKPRB command. Use the parameters to display a specific problem or group of problems rather than the entire list.

```

Work with Problem (WRKPRB)

Type choices, press Enter.

Problem identifier . . . . . *ALL      Character value, *ALL
Status type . . . . . *ALL      *ALL, *OPENED, *READY...
      + for more values
Severity . . . . . *ALL      *ALL, 1, 2, 3, 4
      + for more values

Period:
Start time and date:
Start time . . . . . *AVAIL     Time, *AVAIL
Start date . . . . . *BEGIN     Date, *BEGIN, *CURRENT
End time and date:
End time . . . . . *AVAIL     Time, *AVAIL
End date . . . . . *END       Date, *END, *CURRENT

Hardware:
Device type . . . . . *ALL     Character value, *ALL
Model number . . . . . *ALL     Character value, *ALL
Serial number . . . . . *ALL     Character value, *ALL
Resource name . . . . . *ALL     Name, *ALL
More...

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

```

```

Work with Problem (WRKPRB)

Type choices, press Enter.

Resource name . . . . . *ALL     Name, *ALL

Product:
Program identifier . . . . . *ALL  Character value, *ALL
Release . . . . . *ALL          Character value, *ALL
Modification level . . . . . *ALL  Character value, *ALL
Function . . . . . *ALL

Program . . . . . *ALL          Name, generic*, *ALL
Message identifier . . . . . *ALL  Name, generic*, *ALL

Origin:
Network identifier . . . . . *ALL  Name, *ALL, *NETATR
Control point name . . . . . *ALL  Name, *ALL, *NETATR
Service number . . . . . *ALL     Character value, *ALL
User assigned . . . . . *ALL     Name, *ALL
Group assigned . . . . . *ALL     Character value, *ALL
Problem type . . . . . *ALL     *ALL, 1, 2, 3, 4, 5, 6
Problem category . . . . . *ALL   *ALL, *REPORT, *CRITICAL ...
Bottom

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

```

Work with Problem command examples

The following examples illustrate the use of the WRKPRB command with several combinations of parameters used to selectively display the list of problems in the problem log. All of these examples are acceptable for the Display Problems (DSPPRB) command as well. For the DSPPRB command, you could include any acceptable combination of the OUTPUT, OUTFILE, OUTMBR, and TYPE parameters.

To show the Work with Problems display with a list of only those problem records with a status of OPENED or READY that identify a failing device of type 9347, type:

```
WRKPRB STATUS(*OPENED *READY) HDW(9347)
```

If you want to show the Work with Problems display with a list of all problem records created in the problem log for the specific service requester in Figure 2 on page 9, type:


```
WRKPRB  ORIGIN(CSTNET1 CHICAGO1)
```

To display a list containing only the problem records for some specified hardware that has been assigned a severity of High or Medium while preparing the problem for reporting, type:

```
WRKPRB  SEV(1 2)  HARDWARE(9347 001 '10-75234')
```

To display a list containing problem records that have been opened in the period from midnight to noon of the current day, have not yet been analyzed, and identify a specified licensed program ID and program name as the cause of the failure, type:

```
WRKPRB  STATUS(*OPENED) PERIOD((*AVAIL *CURRENT) (120000 *CURRENT))  
LICPGM(XXXXSS1 03 00) PGM(QNOPGM)
```

To display a list of problem records containing machine-detected problems that were opened because message CPF6788 was sent to the system operator message queue and for which problem analysis ran and determined the resource name of the device involved with the failure is TAP01, type:

```
WRKPRB  RESOURCE(TAP01) MSGID(CPF6788)
```

The list of problems shown by the example above also includes user-detected problems for which you specified the resource name and message ID using the Analyze Problem (ANZPRB) command.

To display a list of problem records containing only problems that have been reported to IBM service support (*IBMSRV) and have been assigned a specified service-assigned identifier by IBM service support (*IBMSRV), type:

```
WRKPRB  SRVID(12345)
```

Display Problems (DSPPRB) command

The Display Problems (DSPPRB) command allows you to display, print, or create an output file for a specific problem record or group of problem records and the corresponding problem information.

Output files allow you to write control language (CL) and other high-level language programs specifically for your environment. The following list provides you with ideas for programs you could write using the problem log output files.

- Create a command to create a tape holding all the PTFs requested for a particular problem ID.
- Create a program to list all remote PTF orders that are not closed, but for which all PTFs are available. You could then send a response to the associated service requesters and the PTF orders can be closed.
- Create a program to delete the problem records for all closed PTF orders (or service requests).
- Create a program to create a report listing all network maintenance activities for a specified period of time.

The parameters for the Display Problems (DSPPRB) command are the same as the parameters for the Work with Problem (WRKPRB) command described previously with some additional parameters. The Display Problems prompt display is shown below with default values for all the parameters listed, except *OUTFILE has been specified for Output.

```

Display Problems (DSPPRB)

Type choices, press Enter.

Problem identifier . . . . . *ALL      Character value, *ALL
Status type . . . . . *ALL      *ALL, *OPENED, *READY...
      + for more values
Severity . . . . . *ALL      *ALL, 1, 2, 3, 4
      + for more values

Period:
  Start time and date:
    Start time . . . . . *AVAIL    Time, *AVAIL
    Start date . . . . . *BEGIN    Date, *BEGIN, *CURRENT
  End time and date:
    End time . . . . . *AVAIL    Time, *AVAIL
    End date . . . . . *END      Date, *END, *CURRENT

Hardware:
  Device type . . . . . *ALL      Character value, *ALL
  Model number . . . . . *ALL     Character value, *ALL
  Serial number . . . . . *ALL     Character value, *ALL
                                          More...

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

```

To show all the parameters for this command, you must specify *OUTFILE for the output parameter.

```

Display Problems (DSPPRB)

Type choices, press Enter.

Resource name . . . . . *ALL      Name, *ALL
Licensed program:
  Program identifier . . . . . *ALL  Character value, *ALL
  Release . . . . . *ALL          Character value, *ALL
  Modification level . . . . . *ALL  Character value, *ALL
Level . . . . . *ALL            Character value, *ALL
Function . . . . . *ALL
Program . . . . . *ALL          Name, generic*, *ALL
Message identifier . . . . . *ALL   Name, generic*, *ALL
Origin:
  Network identifier . . . . . *ALL  Name, *ALL, *NETATR
  Control point name . . . . . *ALL  Name, *ALL, *NETATR
Service number . . . . . *ALL      Character value, *ALL
User assigned . . . . . *ALL       Name, *ALL
Group assigned . . . . . *ALL      Character value, *ALL
Problem type . . . . . *ALL       *ALL, 1, 2, 3, 4, 5, 6
Problem category . . . . . *ALL    *ALL, *REPORT,*CRITICAL ...
                                          More...

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

```

```

                                Display Problems (DSPPRB)

Type choices, press Enter.

Output . . . . . > *OUTFILE      *, *PRINT, *OUTFILE
File to receive output . . . . . FILEA      Name
Library . . . . . *LIBL          Name, *LIBL, *CURLIB
Output member options:
  Member to receive output . . . *FIRST      Name, *FIRST
  Replace or add records . . . . *REPLACE *REPLACE, *ADD
Type of information . . . . . *BASIC      *BASIC, *CAUSE, *FIX ...

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

```

Change Problem (CHGPRB) command

The Change Problem (CHGPRB) command allows you to change the text that appears in some of the fields normally filled in by the system when the problem handling functions are used. The fields you can change are the problem severity, the user assigned to work with the problem, the group to which the problem is assigned, the IBM service number, and the description of the problem. The Change Problem prompt display is shown below.

```

                                Change Problem (CHGPRB)

Type choices, press Enter.

Problem identifier . . . . . 9002489316  Character value, *ALL
Origin:
  Network identifier . . . . . CSTNET1     Name, *NETATR, *ALL
  Control point name . . . . . CHICAGO1  Name, *NETATR, *ALL
Severity . . . . . 2                    *SAME, *NONE, 1, 2, 3, 4
User assigned. . . . . JANE              Name, *SAME, *NONE, *DEFAULT
Group assigned . . . . . *SAME           Name, *SAME, *NONE
Service number . . . . . 16X98          Character value, *SAME, *NONE
Problem category . . . . . *SAME         *SAME, *REPORT, *LOGONLY
Text 'description' . . . . . 'Your own description of the problem'

Bottom

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

```

Using the Change Problem (CHGPRB) command you can tailor some of the problem information that appears on the Work with Problems display to be more meaningful for your enterprise. For example, you can set the problem severity before you prepare a problem for reporting. You can then use this parameter value later with the WRKPRB or DSPPRB command to work with, or display, only problems with a specific severity.

The user-assigned parameter allows you to indicate that a particular user is assigned to a problem. For example, if you have a programmer in your enterprise who is very good at resolving communications problems, you could use this parameter value to indicate that this is a communications problem and that specific programmer will be working on it. You can also use the value of the user-assigned parameter for the WRKPRB or DSPPRB commands to work with, or display, only problems assigned to a specific user.

Note: The command does not check to see if the assigned user has a valid user profile.

The group-assigned parameter allows you to specify a value to use to group problems in a way meaningful to you. You can let the system assign the group if you are using problem filtering (see “Problem filtering” on page 176).

The service number parameter allows you to type the 5-character service-assigned identifier assigned to the problem by IBM service support. This identifier is added to the problem record automatically when you report the problem to IBM electronically. However, if you cannot report the problem electronically, or you choose to report the problem over the telephone, you can use the Change Problem (CHGPRB) command to add this number to the information in the problem record. You can also use this number as a parameter for the Work with Problem (WRKPRB) or Display Problems (DSPPRB) command to work with, or display, only this particular problem.

The problem description parameter allows you to type your own description of the problem. This description appears on the Work with Problems display and in the problem details. Doing this allows you to customize the appearance of the problem log to meet the needs of your enterprise.

Change Problem command examples

The following examples illustrate the use of the CHGPRB command. Combinations of parameters are used to change values in the problem information.

To add service-assigned number 12X34 to the problem record for problem 9001087456 that occurred on the local system, type:

```
CHGPRB PRBID(9001987456) SRVID(12X34)
```

To change the system-defined problem description for problem 9002489316 that occurred on system CHICAGO1, in network CSTNET1, type:

```
CHGPRB PRBID(9002489316) ORIGIN(CSTNET1 CHICAGO1)
TEXT('Printer 10 failed at main site')
```

Delete Problem (DLTPRB) command

The Delete Problem (DLTPRB) command allows you to delete problem records from the problem log. The following display shows the default values for the Delete Problem (DLTPRB) command.

```

Delete Problem (DLTPRB)

Type choices, press Enter.

Problem identifier . . . . . *ALL      Character value, *ALL
Status type . . . . . *ALL      *ALL, *OPENED, *READY...
      + for more values
Days . . . . . 30              0-999
Origin:
  Network identifier . . . . . *NETATR  Name, *NETATR, *ALL
  Control point name . . . . . *NETATR  Name, *NETATR, *ALL

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

```

To delete problems having any status other than CLOSED, the problems must be OPENED longer than the number of days specified by the system value for problem hold interval (QPRBHLDITV).

To delete problems with a status of CLOSED, the problems must be CLOSED longer than the number of days specified by the system value, QPRBHLDITV.

You can set this system value using the Change System Value (CHGSYSVAL) command. The default value for QPRBHLDITV is 30 days. IBM service support (*IBMSRV) recommends keeping problem data for 30 days.

The Delete Problem (DLTPRB) command also has parameters that allow you to specify a problem or group of problems that you want to delete. For example, if you want to delete all problems added to the problem log earlier than 15 days ago and have a status of OPENED, READY, or SENT, type:

```
DLTPRB STATUS(*OPENED *READY *SENT) DAYS(15)
```

Problem records should be deleted on a regular basis to conserve storage space. When a problem record is first created it uses approximately 2412 bytes of disk space. An average problem record containing the data in the following list can use 3600 bytes of disk space:

- Problem information
- 4 events logged
- 2 lines of notes
- Reporting information
- PTF information
- Possible cause information

When you delete problem records from the problem log, you need to reorganize all the physical files used by the problem log functions. You can have the system do this automatically by setting up the cleanup options available with the Operational Assistant function. You can use the automatic cleanup option to delete old messages, printed output, journals, system logs, and calendar items according to the schedule you specify. Cleanup is performed to get rid of objects or files that are no longer needed and are taking up space on the system. To set up your system cleanup options, do the following:

1. Type GO MANAGESYS on any command line and press the Enter key. The Manage Your System, Users, and Devices menu is shown.
2. Select option 60 (Customize your system, users, and devices) and press the Enter key.
3. Select option 2 (Cleanup tasks) and press the Enter key. The Cleanup Tasks menu is shown.
4. Select option 1 (Change cleanup options) and press the Enter key. The Change Cleanup Options display is shown.
On this display, you can view or change the cleanup options for your system. You can have the system cleanup begin manually or automatically and set the time you would like the cleanup to begin. You can also specify the number of days for which system information is to be kept on the system before it is cleaned up.
5. Type a number from 1 to 366 in the *System journals and system logs* field. This is the field that controls the cleanup of the problem log files and associated journal.
6. Type the values you want for system cleanup in the remaining fields and press the Enter key. The cleanup options for your system are now set up.

Once you have set up the cleanup options, and you have allowed automatic cleanup to take place, the system takes care of the rest.

Working with a problem

To work with a particular problem, use the Work with Problem (WRKPRB) command to display the problem log (or use the parameters to look at a specific problem or group of problems), and type an 8 (Work with problem) in the *Opt* column of the problem with which you want to work. The Work with Problem menu is displayed. From this display you can:

- Analyze the problem
- Report the problem
- Run recovery procedures
- Verify the problem has been corrected
- Answer the problem
- Send a response to the service requester who sent a service request
- Close the problem
- Collect, display, save, and delete APAR data

Some of the options listed above are displayed only when the problem record is in a particular status. The explanations below describe the operations you can perform and the conditions under which the operations are available.

Analyzing a problem

The Analyze problem option allows you to run problem analysis procedures. If you are at a service provider, you can run problem analysis directly on the system where the problem occurred.

If the problem you are working with is user-detected, you cannot run problem analysis more than once using the same problem record to store the results. You can run problem analysis again using either the Analyze Problem (ANZPRB) command or the function key provided on the Work with Problems display; however, doing this creates another problem record in which the results are stored.

The original problem record remains unchanged. Problem records created by using the ANZPRB command are considered ready to be REPORTED at once and do not show the Analyze option.

Reporting a problem

The Report problem option is displayed only if the operator is authorized to the Send Service Request (SNDSRVRQS) command. Use this option to enter information necessary for a service request. The status of the problem is then changed to PREPARED. You can then continue and send the service request to the next-level service provider. If the problem has already been prepared, you are shown the Verify Contact Information display again with any prior changes shown. The problem status is then changed to SENT when the request is acknowledged by the next-level service provider.

Note: Problems with a status of OPENED can be reported to an iSeries service provider, but not to IBM service support (*IBMSRV).

Running recovery procedures for a problem

The Run recovery option is displayed only if recovery procedures are available and the problem is not in OPENED status. Recovery procedures are usually related to equipment or media failure. These procedures may produce a possible causes list separate from the one associated with problem analysis.

Verifying a problem is corrected

The Verify problem corrected option is displayed only if the problem is not in OPENED status. This option allows you to run verification procedures if they are available. If verification procedures are not available, you can confirm the problem has been solved by contacting the operator at the service requester, and then selecting the option for indicating the problem has been corrected.

Answering a problem

The Answer problem and the Send response options are only available on an OS/400 service provider.

Selecting the Answer problem option presents a menu with options for having the system search for PTFs to solve the problem or for working with PTF information associated with the problem record.

When you select the option to answer the problem from the PTF database, it allows the system to search for solutions to the problem and to write the results of the search into the problem record. If no PTFs are found that match the symptoms of the problem, no PTF numbers are put into the problem log and a message is shown indicating that no PTFs were found.

When you select the option to work with PTF information, a display is provided from which you can add or remove PTF numbers associated with the problem. From this display you can also look at the cover letters associated with the PTFs when the cover letters are on the system. For more information about answering problems, see "Answering problems" on page 188.

Note: The Answer problem option does not appear for problem records created as a result of local PTF orders.

Sending a response for a problem

The Send response option is only displayed when the problem originated on another system in the network and its status is ANSWERED or VERIFIED. This option sends a response to the service requester that reported the problem and updates the problem record on that system with answer information. Answer information may be a list of possible causes or a list of PTFs that solve the problem. The Send response option also sends the PTFs to the service requester. If the PTFs are not available on your system (no save files exist or no PTFs are on hold), you cannot send a response.

Notes:

1. If you want to distribute a PTF and the distribution queue value is *SNADS in the Work with Service Requesters (WRKSRVRQS) command, you must have *USE authority to the Send PTF (SNDPTF) command.
2. If you want to distribute a PTF and the distribution queue is *SVDS in the Work with Service Requesters (WRKSRVRQS) command, you must have *USE authority to the CPYPTF command.

Closing a problem

The Close problem option is available on all iSeries servers and allows you to close problems on which work is complete. If you close a problem, the Work with Problem option is no longer available and you can no longer perform any update operations for that problem.

Collecting authorized program analysis report (APAR) data

When you select this option, the information is collected and then stored in a library.

The Save APAR data to APAR library option is available only if:

- The user is authorized to the Save APAR Data (SAVAPARDTA) command
- The problem entry is not a PTF order
- The problem was not received from a service requester

This option causes the SAVAPARDTA command to be run for the problem specified.

When you select this option, the information is collected and then stored in a library. If you select this option for a problem that occurred on a remote system, the associated spooled files and the error log information will not be available. In this case, you should contact an operator at the service requester and have them save the APAR data.

An **authorized program analysis report (APAR)** is a request for the correction of a defect in a current release of an IBM-supplied program. It contains information about the problem from the problem record, any associated spooled files, the error log entry, and any available vital product data available for the licensed program. **Vital product data** is a structured description of a device or program. In this case, the vital product data includes:

- The name of the licensed program, Licensed Internal Code Group, or product
- The release and modification, and the program module names
- The national languages selected
- Other information may be included about the failing program.

Displaying authorized program analysis report (APAR) data

The Display APAR library option is available only if the problem entry has APAR data associated with it (the APAR library is identified).

This option causes the Display Library (DSPLIB) command to be run, specifying the APAR data library name.

Save authorized program analysis report (APAR) data

The Save APAR library option is available only if:

- The user is authorized to the Save Library (SAVLIB) command
- The problem entry has APAR data associated with it (the APAR library is identified)

This option causes the SAVLIB command prompt to be displayed with the library parameter filled in.

When you select this option, the APAR library associated with the specified problem is stored on a save tape you select or in a save file. You can then mail the APAR data stored on the tape to your IBM service support representative to have the data analyzed. This data can also be sent to a service provider, where the data can be restored using the Restore APAR Data (RSTAPARDTA) command.

Delete authorized program analysis report (APAR) data

The Delete APAR data option is available only if:

- The user is authorized to the Delete APAR Data (DLTAPARDTA) command
- The problem entry has APAR data associated with it (the APAR library is identified)

This option causes the Delete APAR Data (DLTAPARDTA) command to be run, which deletes the APAR data and removes all references to the data from the problem entry.

For more information about analyzing, reporting, and answering problems, see Chapter 9, "Detecting, analyzing, and reporting remote problems" on page 179.

Understanding problem status changes

Problem status helps you keep track of where you are in the process of working with a problem. The following section describes each status. Table 4 on page 174 shows the final status when operations are performed on a problem record.

OPENED status

A problem with a status of OPENED indicates the problem has been detected. More information about the problem can be obtained by running problem analysis from the problem record, or by retrieving information from the system on which the problem originated.

When a record is first created for a system-detected problem, its status is OPENED. A problem record with OPENED status is also created on a service provider as the result of receiving an alert or receiving a service request for a system-detected problem that has not yet had problem analysis performed.

A problem record created as the result of an alert does not contain as much information as a problem record created by sending a service request. If the record is for a system-detected problem, running problem analysis provides the additional data.

A problem in OPENED status can be reported to an iSeries service provider; however, you cannot report a problem in OPENED status to IBM service support (*IBMSRV). Figure 20 shows the operations available to both a service requester and a service provider when a problem is in OPENED status.

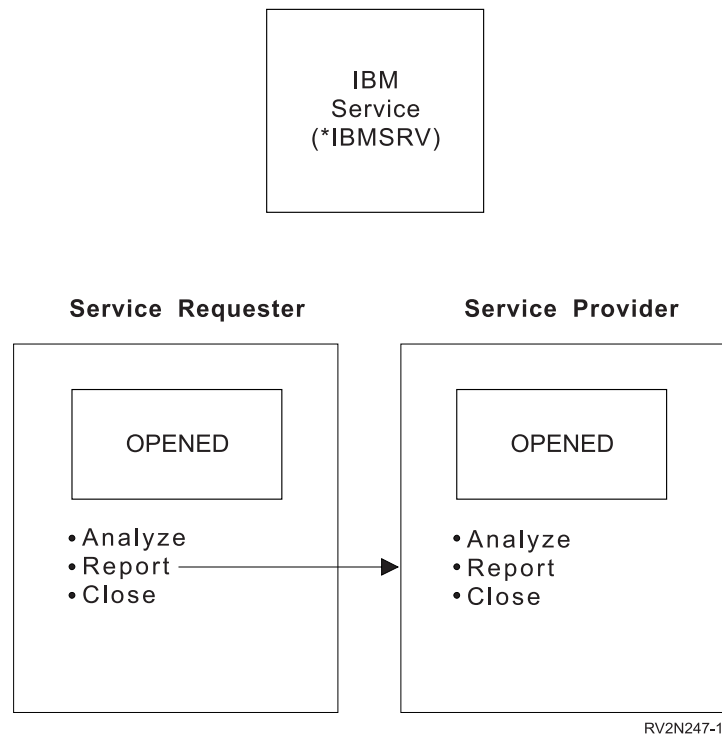


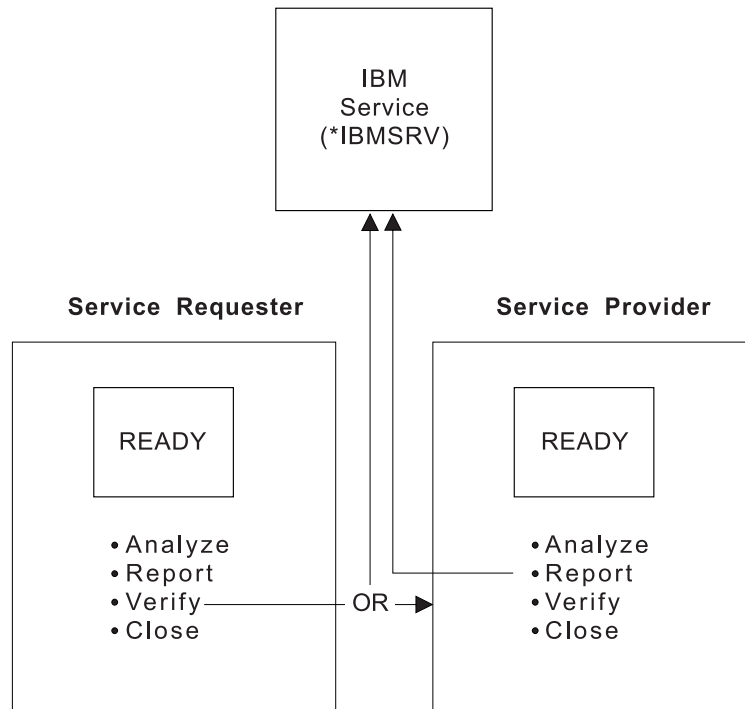
Figure 20. Operations Available for a Problem in OPENED Status

READY status

READY status indicates that problem analysis information has been added to the problem record so that it may be reported.

A problem record is created at READY status when the problem record is created using the Analyze Problem (ANZPRB) command. An OPENED problem record changes to READY status when problem analysis is run on the problem. A problem record with PREPARED status changes back to READY by running problem analysis again. A problem record with a status of READY is created on a service provider when a problem record that has been at READY status is reported to the service provider by way of a service request.

Figure 21 on page 169 shows the operations available to both a service requester and a service provider when a problem is in READY status.



RV2N133-2

Figure 21. Operations Available for a Problem in READY Status

PREPARED status

A problem record with a status of PREPARED indicates the problem record has had information added that is necessary to report to the next level service provider. You can prepare a problem for reporting by selecting option 2 (Report problem) from the Work with Problem menu. You can then use the Send Service Request (SNDSRVRQS) command to send all problems in PREPARED status to the next level service provider.

The status of a problem can change to PREPARED from any other status except CLOSED by selecting option 2 (Report problem) on the Work with Problem menu. Figure 22 on page 170 shows the operations available to both a service requester and a service provider when a problem is in PREPARED status.

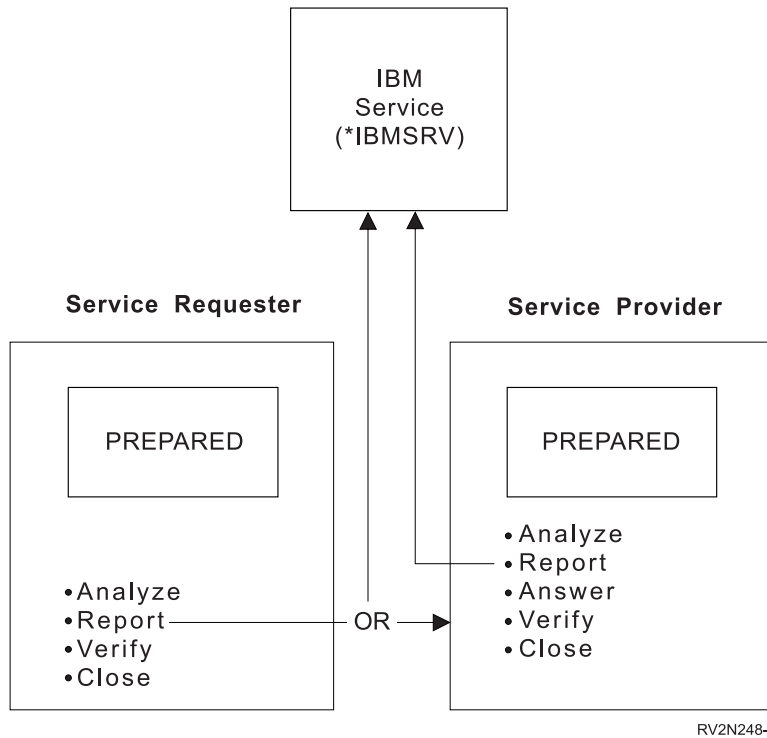


Figure 22. Operations Available for a Problem in PREPARED Status

SENT status

SENT status indicates a service request has been sent from a service requester and received by a service provider; however, information necessary to correct the problem has not been sent to the service requester. An acknowledgement is sent by the service provider when the request is received.

The status of a problem record is changed to SENT when option 2 (Report problem) is selected from the Work with Problem menu, or when the function key to report a problem is pressed from the Problem Analysis Complete display. The problem status also changes to SENT when the Send Service Request (SNDSRVRQS) command is run, and the option is selected to send the service request now or to send the service request by voice. On a service requester, a problem record may also change to SENT status after remote problem analysis is complete.

Figure 23 on page 171 shows the operations available to both a service requester and a service provider when a problem is in SENT status.

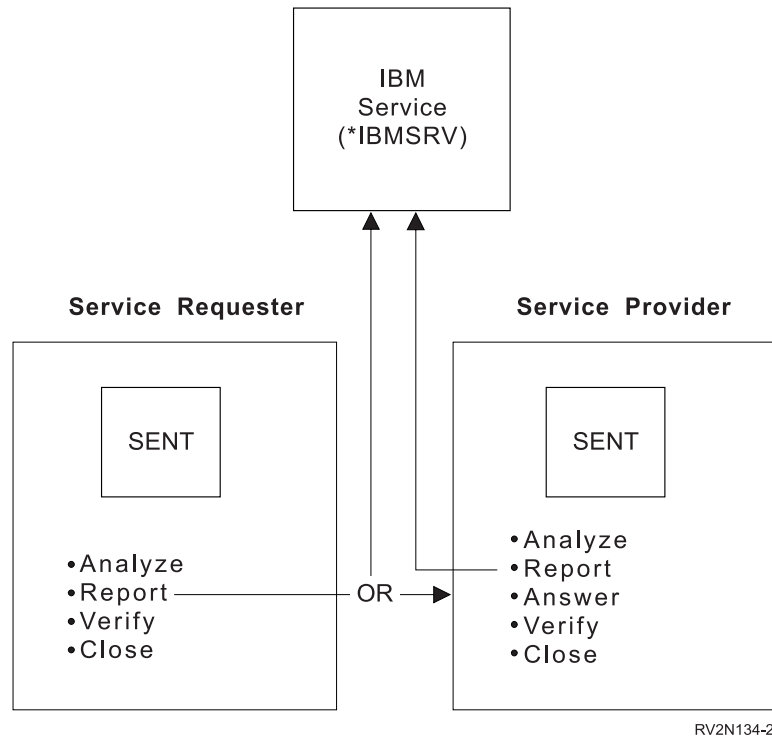


Figure 23. Operations Available for a Problem in SENT Status

ANSWERED status

ANSWERED status indicates that an answer to the service request providing information to correct the problem has been received, or that answer information has been added to the problem record locally. If the answer involves PTFs, a list of PTFs is available. The actual PTFs may or may not be sent with the PTF list.

The status of a problem record is changed to ANSWERED when option 2 (Report problem) is selected from the Work with Problem menu at the service requester and when an answer is returned by the service provider. The answer for a problem may consist of a PTF list with or without the actual PTFs, or it may consist of a list of possible causes.

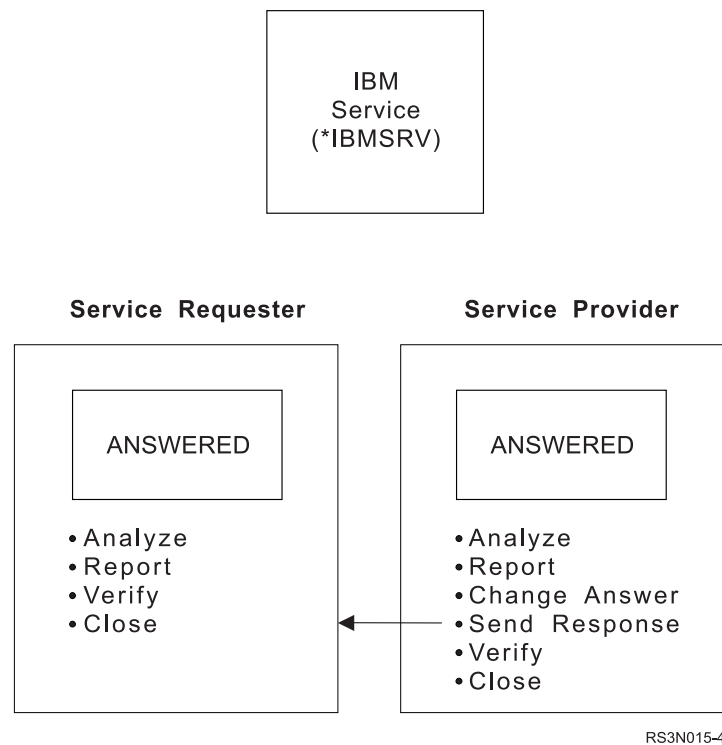
When the problem is received at the service requester, the problem is answered by selecting the option to answer the problem from the Work with Problem menu.

The status of a problem also changes to ANSWERED when you select one of the two options available on the Answer Problem display. Option 1 (Answer from PTF database) allows you to search the local PTF database on the service provider. Option 2 (Work with PTF information) allows you to add PTF information to the problem record.

You can view the PTF numbers and their status on the service provider by selecting the option to answer the problem from the Work with Problem menu, and then selecting the option to Work with PTF information. The list of PTFs is shown indicating whether they are held or released.

Note: Problems may be answered automatically, depending on the way you have set up your service provider. If you are answering the problem manually, you need to select the option to send a response to the problem for the PTFs to be sent to the service requester.

When the problem record is in ANSWERED status, it may be used to send a response to the service requester. Figure 24 shows the operations available to both a service requester and a service provider when a problem is in ANSWERED status.



RS3N015-4

Figure 24. Operations Available for a Problem in ANSWERED Status

VERIFIED status

VERIFIED status indicates that the operator has verified that the problem is corrected.

You can change the status of a problem record to VERIFIED by selecting option 4 (Verify problem corrected) from the Work with Problem menu.

You can use a problem record with VERIFIED status to send a response to the service requester.

Figure 25 on page 173 shows the operations available to both a service requester and a service provider when a problem is in VERIFIED status.

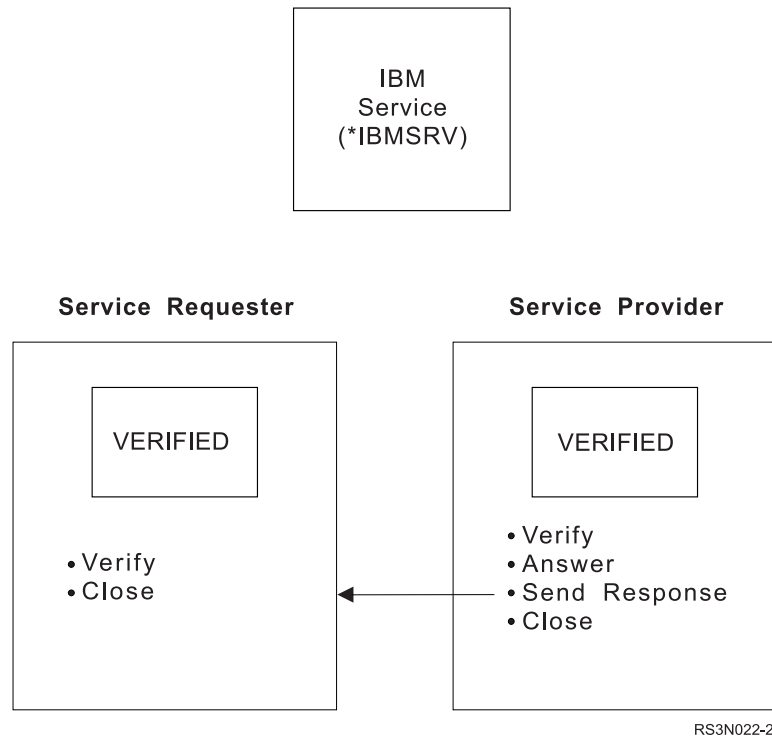


Figure 25. Operations Available for a Problem in VERIFIED Status

CLOSED status

CLOSED status indicates that problem tracking is complete for this problem. All the information pertaining to this problem record can still be viewed, but no other operations or status changes are allowed.

Any problem record with any status may be changed to CLOSED by selecting the Close problem option from the Work with Problem menu.

Status changes and operations

Table 4 summarizes the status changes and related operations for a problem that has been recorded in the service provider problem log. All operations shown in the table are available at the service provider except for system detection and PTFs ordered. System detection creates a problem record on the system where the problem originated. Ordering PTFs creates a problem record on the system from which the PTFs were ordered.

When a service provider reports a problem, the status may be either SENT or ANSWERED. This depends on whether an answer is returned immediately from the next level service provider.

Note: Not all operations available for problem records result in a change of status.

Table 4. Problem Record Status Changes Based on Operations Performed

Operations	Starting Status							
	No Record	Opened	Ready	Prepared	Sent	Answered	Verified	Closed
System detects problem	O							
Run ANZPRB (L)	R							
Run ANZPRB (R)	R							
PTF ordered	R							
Analyze problem (L)		R	R	R	S	A	V	
Analyze problem (R)		R	R	R	R	R	R	
Report (send now)		S	S/A	S/A	S/A	S/A	S/A	
Report (send later)		P	P	P	P	P	P	
Report (voice)		S	S/A	S/A	S/A	S/A	S/A	
Run recovery			R	P	S	A	V	
Verify corrected			V	V	V	V	V	
Answer problem			A	A	A		A	
Change answer						A		
Send response						A	V	
Close problem		C	C	C	C	C	C	
Collect APAR data		O	R	P	S	A	V	
Display APAR data		O	R	P	S	A	V	
Save APAR data		O	R	P	S	A	V	
Delete APAR data		O	R	P	S	A	V	

Table 4. Problem Record Status Changes Based on Operations Performed (continued)

Operations	Starting Status							
	No Record	Opened	Ready	Prepared	Sent	Answered	Verified	Closed
Note: <ul style="list-style-type: none"> • A = Answered • C = Closed • O = Opened • P = Prepared • R = Ready • S = Sent • V = Verified • Blank = Not Allowed • (L) = Local • (R) = Remote 								

In the service requester problem log, the problem status changes following problem analysis are different. When a user-detected problem occurs and the network operator at the service provider uses the Analyze Problem (ANZPRB) command for remote problem analysis, the status of the problem record created on the service requester is SENT.

When a system-detected problem occurs and the status of the problem on the service requester is OPENED, READY, or PREPARED, the status changes to SENT on the service requester after the problem is analyzed remotely. When you run remote problem analysis on a system-detected problem that is in SENT, ANSWERED, or VERIFIED status on the service requester, the status does not change and the problem record is not updated with the problem analysis results.

You cannot run remote problem analysis on a system-detected problem when the status of the problem on the service requester is CLOSED or when the problem record does not exist. The answer problem, change answer, and send response operations are not available on the service requester.

Tracking problems

One of the tasks you want to do when working with problems from systems in a network is to track the problem. You may want to look at the time the problem first occurred, how many times problem analysis was run, or whether the problem was answered and verified. Checking the history of a problem can keep the operator at a service provider aware of the progress made on resolving the problem and prevent operations, such as problem analysis, from being performed more times than necessary.

When you fix a problem by sending a program temporary fix (PTF), a change request is submitted. You can track the status of the distribution of that PTF by displaying submitted change requests. For more information about displaying submitted change requests, see "Tracking submitted change requests" on page 210.

Problem history can be viewed at both the service provider and the service requester. Although the operators may be looking at the history for the same problem, the events recorded are different. This is because the events occurred on different systems. For certain events, the history is recorded on both systems

concurrently. These events are running remote problem analysis, remote verification procedures (when available), and remote recovery procedures (when available).

To display the history of a problem from the time the problem record was created by the system to the current date and time, do the following:

1. Type the Work with Problem (WRKPRB) command or the Display Problems (DSPPRB) command on any command line and press the Enter key. If you can limit the list of problems using the parameters of these commands, it may help you find the problem you are looking for more easily.
2. Type the option number to display the problem details in the option column of the problem for which you want to display the history, and press the Enter key. The Display Problem Details display is shown with detailed information about the problem you selected.
3. Press F6 (Display problem history). The Display Problem History display is shown.

The following display is an example of the Display Problem History display.

```
Display Problem History

Problem ID . . . . . : 9008140122
Origin . . . . . : CSTNET1.CHICAG01
Current status . . . . . : ANSWERED
Problem . . . . . : User detected a software problem on a different
                   AS/400.

Date      Time      User ID   Event
03/22/90  11:41:10  OPR      Problem entry opened
03/22/90  11:41:21  OPR      Prepared to report
03/22/90  11:41:51  OPR      Service request sent
03/22/90  11:42:02  OPR      Problem answered

Press Enter to continue.

F3=Exit  F12=Cancel

Bottom
```

Problem filtering

Problem filtering is provided as a function of the problem log manager, which is part of the OS/400 program. It is started when a problem is created, changed, or deleted. A problem filter must be created and specified in the system value QPRBFTR for filtering to occur.

Problem filtering enables the grouping of problems into logical, manageable sets. Each problem group has specific tasks or actions assigned to it for resolving the problem. After a problem is assigned to a group, the specified actions are automatically performed, such as assigning the problem to a specific user ID or routing a notification record to a specific data queue. You can use an application program to monitor the data queues used by filtering.

You can develop automatic problem handling applications for resolving problems by using such commands as Change Problem (CHGPRB), Display Problem (DSPPRB), and Delete Problem (DLTPRB).

The commands for working with problem filters are:

- Work with Filters (WRKFTR)
- Create Filter (CRTFTR)
- Change Filter (CHGFTR)
- Delete Filter (DLTFTR)
- Work with Filter Selection Entry (WRKFTRSLTE)
- Remove Filter Selection Entry (RMVFTRSLTE)
- Add Problem Selection Entry (ADDPBBSLTE)
- Change Problem Selection Entry (CHGPRBSLTE)
- Add Problem Action Entry (ADDPBACNE)
- Change Problem Action Entry (CHGPRBACNE)

The Work with Problem (WRKPRB) command also added an additional parameter to subset by group. The group was defined when the problem was filtered.

A **registered filter** allows you to have more than one active filter for alerts and the problem log. When a filter is registered, the system can receive notification of events on a data queue. Each event type can have a registered filter that can be deregistered when the notifications are no longer needed. To register a filter, you need to use application program interfaces (APIs).

Chapter 9. Detecting, analyzing, and reporting remote problems

This chapter assumes that you are familiar with running local problem analysis on your iSeries server and that you are aware of the problem handling procedures described in the Getting Started topic in the iSeries Information Center.

Problem analysis consists of online procedures designed to help you determine the cause of a system problem, gather information about the problem, resolve the problem if possible, and report the problem if necessary. You can analyze two types of problems using iSeries server problem analysis: system-detected problems, which are usually equipment related, and user-detected problems, which are usually job or program related.

Having System Manager installed on one or more iSeries servers in your network allows the service providers to start problem analysis on any service requester that has been entitled to service from that service provider. This allows you to have service requesters in your network that do not have full-time operators available to run local problem analysis.

Running problem analysis remotely gives you the opportunity to resolve system problems in your network more easily. In many cases, problem analysis can identify the cause of a problem and the location of the program or equipment that is failing. Most of the time you can correct the problem yourself without assistance. When remote problem analysis is complete, the service provider can send a service representative to the service requester with the replacement part, or distribute the necessary program temporary fixes (PTFs) immediately to the service requester where the problem occurred. This limits the amount of time spent on service calls and restores normal system operations more quickly. An example of the communications flow for problem management tasks related to a user-detected problem is shown in Figure 26 on page 180.

Note: SYSC in Figure 26 on page 180 can be either an OS/400 service provider or IBM Service, which is designated as *IBMSRV.

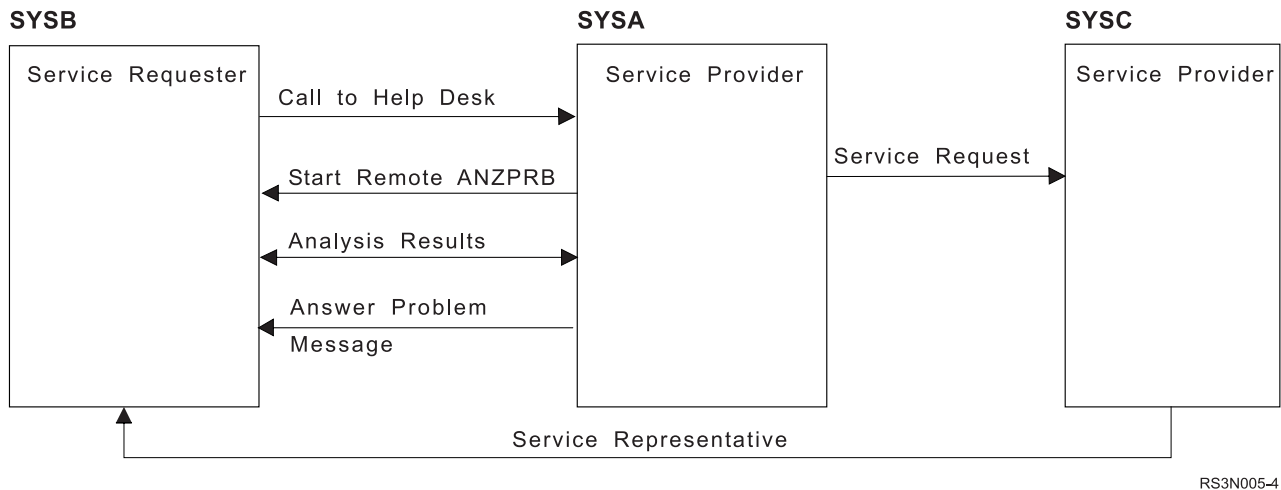


Figure 26. Communications Flow for Problem Management

Detecting problems

There are two types of problems you can analyze using OS/400 problem analysis. System-detected problems are usually related to an equipment failure or a program error. User-detected problems are usually related to a job or program problem.

System-detected problems

The iSeries server and its attached devices are able to detect some types of problems. When a problem is detected, several operations take place:

- An error log entry is created
- A problem record is created
- A message is sent to the QSYSOPR message queue
- An alert may be created

Information is recorded in the error log and the problem record. The alert is then sent to the service provider if the service provider is either an alert focal point or the network node server for the system with the problem. The error log and the problem record may contain the following information:

- Vital product data
- Configuration information
- Reference code
- Name of associated device
- Additional failure information

Once the problem has been detected, you can start problem analysis.

Application-detected problems

Your application can detect some types of problems such as unexpected return codes. An application can use the First Failure Data Capture (FFDC) function of the OS/400 program to:

- Create a problem record
- Send a message to the QSYSOPR message queue

- Send an alert
- Collect data related to the problem
- Build a symptom string

This problem record is created only if FFDC is running without threads support. The FFDC function does change depending on the thread support used.

User-detected problems

User-detected problems are usually related to program errors that may cause any of the following problems to occur:

- Job problems
- Incorrect output
- Messages indicating a program failure
- Device failure not detected by the system
- Poor performance

When a user detects a problem, no information is gathered by the system until problem analysis is run, or you select the option to save information to help resolve a problem from the Operational Assistant USERHELP menu.

When you choose to start problem analysis for a problem, the analysis can be done either locally at the service requester or remotely from the service provider using the Analyze Problem (ANZPRB) command. When you use the Analyze Problem (ANZPRB) command, a series of displays is presented that allows the system to collect information about the problem. You can then select an option to create a problem record and report the problem. When problem analysis is complete, the information collected is added to the problem record. If problem analysis is run remotely from the service provider, problem records are created and updated on both systems.

When a user selects the Operational Assistant option to save information to help resolve a problem, information about that user's job and sign-on session is saved in spooled files that are associated with a problem record. The service representative at the service provider can either instruct the user to send the spooled files and the problem record to the service provider or sign on to the remote service requester and look at the spooled files to determine whether the problem is with an application program or with an IBM licensed program. If the problem is occurring in an IBM licensed program, the service representative should instruct the user at the service requester to select the option to work with the problem from the Work with Problems display, and then select the option to save authorized program analysis report (APAR) data from the information in the problem record (see "Collecting authorized program analysis report (APAR) data" on page 166). The APAR information can then be sent to IBM service support for IBM products.

Receiving problem notifications

A problem notification can be a telephone call, a service request, or an alert from a service requester to a service provider.

When a problem is detected by an operator or user, there is no method to automatically create a problem record for it. An operator or user at the service requester may notify you about such a problem with a system message or a telephone call.

When a service request or an alert is received that creates a problem record, a message is sent to the message queue you defined when you set up your service provider. This message lets you know a problem has occurred on a service requester. Depending on the values you choose for your service provider attributes, a problem record may be created in the problem log when a problem notification is received.

A problem record is created in the problem log if a service requester:

- Reports a problem to a service provider by way of a service request and the log remote problems (LOGRMTPRB) service provider attribute is set to *SRVRQS or *ALL.
- Sends an alert to the service provider and the log remote problems (LOGRMTPRB) service provider attribute is set to *ALERT or *ALL. The alert must also have an associated problem record on the service requester. Some alerts are notifications for situations that do not have problem records and do not cause a problem record to be created.

Note: This notification is dependent on setting up OS/400 alert support and the designation of the service provider as the alert focal point or the network server for the service requesters that belong to that service provider.

If the service requester that sent the problem notification is not found in the list of systems for which you provide service, a problem record can be created manually; however, no automatic actions take place. This allows you to determine whether the service requester should be entitled to service. If so, you can use the Work with Service Requesters (WRKSRVRQS) command to add the service requester to the list of systems for which you provide service. You can then work with the problem using the problem record created.

Automatic problem analysis

Specifying *YES for the automatic problem notification parameter on the Change Service Attributes (CHGSRVA) command, you indicate that you want the system to automatically report problems. When you specify *YES, the service provider automatically receives notification of local system problems. On this same command, you can also specify that the system automatically analyze problems at the time of the failure. When you specify *YES, problem analysis routines are run to isolate or correct the problem.

Reporting problems detected by the system

The system **problem log** allows you to display a list of all the problems that have been recorded on the system. You can also display detailed information about a specific problem, such as the product type and serial number of the device that had the problem, the date and time of the problem, the part that failed, where the part is found, and the problem status. You can also analyze and report a problem, or determine any service activity that was done.

To report a problem which has an entry created in the problem log:

1. Type the Work with Problems (WRKPRB) command on any command line and press the Enter key. The Work with Problems (WRKPRB) display appears.
2. If you have a problem ID, look for an entry with the same ID on the Work with Problems display.
3. To change any fields that appear on the Verify Contact Information display, type over the current information and press the Enter key. The new information is included in the service request.

4. Select the severity level that closely relates to the severity of your problem on the Select Problem Severity display.

Note: If a message appears indicating that the problem has not been analyzed, and your attempts at problem analysis fail, then do the following:

- a. Check the message queue to see if there are any messages concerning this problem. Follow the recovery instructions of these messages.
 - b. Call your service representative if previous step does not apply.
5. Select when and how you want to send the service request on the Select Reporting Option display.

For more information about working with problems, see “Work with Problem (WRKPRB) command” on page 157.

Service requests as problem notifications

A service request is sent from a service requester for problems occurring on the system that may or may not have been analyzed.

If an operator at a service requester has run local problem analysis and sent the service request, the results of the analysis are included in the service request. Problem analysis results include: possible causes of the problem, symptoms of the problem, and notes from the operator. When the symptoms are present in a service request, a search for PTFs to answer the problem is done automatically. If all the necessary PTFs are available and the send PTFs automatically (SNDPTFAUTO) service provider attribute is set to *YES, the PTFs are distributed immediately. If a solution is not found, you can send a service request with the problem record to the next level service provider to try to solve the problem.

When the problem has not been analyzed at the service requester, the problem record that is sent is in OPENED status. You can use the problem record to start remote problem analysis on the service requester from the service provider. See “Starting remote problem analysis” on page 184 for a complete description of the methods you can use to start remote problem analysis.

If the problem is user-detected and still occurs, you can use the Analyze Problem (ANZPRB) command to run remote analysis on the service requester.

Note: Running remote problem analysis using the Analyze Problem (ANZPRB) command creates a new problem record on both the service provider and the service requester.

Alerts as problem notifications

When OS/400 alert support is set up, alerts are sent from a service requester automatically when a system-detected problem occurs. The alerts are sent to the service provider either when the service provider is designated as the alert focal point for the service requester or when the service requester is an end node and the service provider is designated as its network node server. Alerts are notifications for system-detected problems only. Problem records created from alerts contain less information than problem records created from service requests until problem analysis is performed.

Alerts provide notification of system-detected problems that may not have been detected by the operator at the service requester. They may provide the only problem notification from a service requester that does not have an operator (alert

status network attribute is *UNATTEND). The problem record created from the alert allows you to run remote problem analysis on service requesters in your network that are not maintained by an operator.

If you want to run remote problem analysis on a problem related to an alert, you need to be sure a problem record was created from the alert on the service provider. The problem record is created either automatically when the alert is received, or manually using the Work with problem option on the Work with Alerts display. Start problem analysis by selecting the option to analyze the problem from the Work with Problem menu, rather than using the Analyze Problem (ANZPRB) command. You want to use this method to start problem analysis because alerts are notifications for system-detected problems. Problem analysis procedures for system-detected problems provide more accurate results for this type of problem than results provided by user-detected problem analysis.

Running remote problem analysis

After a problem has been detected and you are ready to run remote problem analysis, you need to know a few things.

First, to run remote problem analysis for a *system-detected* problem, a problem record must exist on both the service provider and the service requester. To run remote problem analysis for a *user-detected* problem, a problem record need not exist on either system. You can use the Analyze Problem (ANZPRB) command to run remote problem analysis for user-detected problems, which creates new problem records in the problem logs on both systems. The analysis results are stored in both of the problem records.

Another consideration is whether or not an operator is available at the service requester to assist you with physical tasks as you run problem analysis from the service provider. Depending on the type of problem occurring on the service requester, the network operator at the service provider may need assistance. This is usually the case for an equipment or communications failure, although assistance may be required for some user-detected problems as well. Before you begin remote problem analysis, you should check to see whether an operator is available at the service requester to assist you with the analysis.

Starting remote problem analysis

There are several ways to start remote problem analysis on the service requester from the service provider. When you start remote problem analysis, a display station pass-through session is automatically started, allowing you to sign on to a service requester from a service provider and proceed directly to problem analysis procedures on the system with the problem.

If you are performing problem management from a System/370 or a System/390 host system, you can use the host command function to sign on to an OS/400 service provider and then use any one of the methods for starting remote problem analysis. The following sections briefly describe the different ways of starting remote problem analysis.

For system-detected problems

A problem record must exist in the problem logs of both the service requester and the service provider before remote problem analysis can be run for a system-detected problem. The problem record on the service requester cannot be in CLOSED status or deleted; otherwise, an error message is displayed.

From the service provider message queue:

1. Use the Display Messages (DSPMSG) command to display the service provider message queue. (This is the message queue you specified either when you changed the service provider attributes using the Change Service Provider Attributes (CHGSRVPVDA) command, or when you set up your list of service requesters using the Work with Service Requesters (WRKSRVRQS) command.) Messages involving system problems, both local and remote, are indicated by an asterisk (*) in the first column of your display.
2. Move your cursor to the message for a remote problem notification and press F14 (Run problem analysis) to show the Work with Problem display.
3. Type an 8 (Work with problem) in the *Opt* column and press the Enter key.
4. Select option 1 (Analyze problem) from the Work with Problem menu.

From the Work with Problems display:

1. Type the Work with Problem (WRKPRB) command on any command line and press the Enter key. Use the parameters to selectively display a problem or locate the problem using the position to function available at the top of the display.
2. Type an 8 (Work with problem) in the *Opt* column next to the problem you want to analyze and press the Enter key. The Work with Problem menu is shown.
3. Select option 1 (Analyze problem) and press the Enter key. If this option is not available, problem analysis for system-detected problems cannot be run for this problem. In this case, use one of the procedures to start problem analysis for a user-detected problem.

From the Work with Alerts display:

1. Type the Work with Alerts (WRKALR) command on any command line and press the Enter key.
2. Press F11 (Display user/group) to show the problem IDs associated with the alerts.
3. Type a 9 (Work with problem) in the *Opt* column next to the alert associated with the problem you want to analyze and press the Enter key (or press F18 (Work with problem) to work with the problem log).
4. Type an 8 (Work with problem) in the *Opt* column next to the problem you want to analyze.
5. Select option 1 (Analyze problem) from the Work with Problem menu.

This method of starting remote problem analysis is for system-detected problems that cause alerts to be created at the service requester and sent to the service provider.

When local problem analysis is required for remote problems: When a service provider receives a service request from a service requester, the status of the problem record on the service requester is SENT. If for any reason you need to run problem analysis locally on the service requester, you need to change the status in the problem record back to PREPARED so that when analysis is run locally, the analysis results are updated in the problem log.

For example, a service request arrives at the service provider and the network operator begins remote problem analysis. The analysis is not completed because no one is at the remote system to assist with the physical tasks necessary to properly analyze the problem. The network operator or another representative from the

service provider might then want to go to the remote site and run problem analysis locally to perform the physical tasks necessary to complete problem analysis.

Use the following procedure on the service requester to change the status of the problem back to PREPARED:

1. Type the Work with Problem (WRKPRB) command on any command line and press the Enter key.
2. Type an 8 (Work with problem) in the *Opt* column next to the problem in SENT status and press the Enter key.
3. Select option 2 (Report problem) from the Work with Problem menu and press the Enter key.
4. Press the Enter key to pass the Verify Contact Information display.
5. Select a problem severity and press the Enter key.
6. Type a 1 (Select) in the *Opt* column of the service provider you would normally report this problem to.
7. Select option 2 (Do not send service request) and press the Enter key.

The problem is returned to PREPARED status and you can run local problem analysis.

For user-detected problems

For problems that have never been analyzed but were reported by a service request, you can start problem analysis from the service provider message queue and the Work with Problems display, the same as for system-detected problems. Problem records created using the Operational Assistant function allow you to start analysis from the problem record. Problem records created using the Analyze Problem (ANZPRB) command are always analyzed and do not show the option to run analysis.

On any command line on a display:

1. Type the Analyze Problem (ANZPRB) command and press F4 (Prompt).
2. Type the values for the parameters, including the control point name and network ID for the service requester on which you want to run remote analysis.
3. Press the Enter key.

Note: You can also specify the parameters on the command line when you type the command, or simply type ANZPRB and press the Enter key to follow a menu path that allows you to specify the remote service requester.

This method of starting problem analysis is for user-detected problems such as incorrect output or system job problems.

From the Work with Alerts display:

1. Type the Work with Alerts (WRKALR) command on any command line and press the Enter key.
2. Press F16 (Analyze new problem).
3. Use the menus shown to indicate the problem is occurring on another system in the network.

A problem record is created at the completion of problem analysis.

This method of starting problem analysis is for user-detected problems. There is no connection between any of the alerts shown on the Work with Alerts display and this call to problem analysis.

From the Work with Problems display:

1. Type the Work with Problem (WRKPRB) command on any command line and press the Enter key.
2. Press F14 (Analyze new problem).
3. Use the menus shown to indicate the problem is occurring on another system in the network.

In both cases, a problem record is created on both systems at the completion of problem analysis.

This method of starting problem analysis is for user-detected problems. There is no connection between any existing problem records and this call to problem analysis.

You can start remote problem analysis using any of the methods listed previously. However, if the problem you want to analyze is system-detected, you should be sure to use one of the methods to start remote problem analysis for system-detected problems to get the best results. As previously stated, problem analysis procedures for system-detected problems are different from those for user-detected problems. You obtain the best results from problem analysis if you use the appropriate analysis for the problem.

Completing problem analysis

When remote problem analysis procedures are complete, the results are either a possible causes list, which is shown at the service provider, or a symptom string, which is added to the problem details in the corresponding problem record. When the results are in a possible causes list, the symptom string information is also added to the corresponding problem record in the problem logs on both the service provider and the service requester.

Possible causes lists

There are several types of possible causes lists. The type of list shown when problem analysis is complete is dependent on how successful problem analysis procedures are in isolating the problem. The following types of lists indicate the results of running problem analysis.

- **Point of Failure**

This list is presented when problem analysis ends unexpectedly or before it is complete. This list is associated with a system reference code (SRC) and is usually very general. Some examples of when this list might be shown include:

- The operator exits problem analysis by pressing F3 (Exit) before the procedures are complete.
- An error occurs during problem analysis because the corresponding error log entry can not be found.
- The problem analysis procedures necessary require an operator for physical tasks and none is available.

- **Partial Isolation**

This list is presented when problem analysis is lacking a resource to completely isolate the problem, but has enough information to eliminate some of the point-of-failure possible causes.

- **Isolation**

This list is presented when every step in the problem analysis procedure is complete. Problem analysis results are the best possible.

- **Recovery**

This list is presented when the option to run recovery procedures is selected from the Work with Problem menu and a problem is encountered while the recovery procedures are being run.

- **Verification**

This list is presented when the option to run verification procedures is selected from the Work with Problem menu and a problem is encountered while the verification procedures are being run. Generally, this list indicates that whatever was done to correct the problem did not work.

- **Answer**

This list is presented **after** a service request has been sent to IBM service support (*IBMSRV) and the problem information has been checked to determine whether the possible causes are correct for the symptom string created. If the possible causes are not correct, the answer list is sent to the iSeries server and put in the problem log. If the possible causes are correct, a response is sent to the iSeries server indicating that fact. In either case, a service representative is sent if the problem is an equipment failure.

Note: There is no guarantee that the results you obtain from remote problem analysis are any more accurate than results obtained by an operator at the service requester. You may want to run problem analysis again, only if a problem occurs during the first time problem analysis was run.

The result of problem analysis is a symptom string that automatically begins the system's search of the PTF database. If a match is found, the numbers of the necessary PTFs are written into the problem record and may or may not be distributed immediately, depending on your service provider attributes and the availability of the PTFs on your system. If no match is found, you need to send a service request to your next level service provider or to IBM service support (*IBMSRV).

When the problem analysis result is a possible causes list that shows the highest probability possible cause is a piece of equipment or a part, a service representative should be sent with the replacement part to the service requester where the problem occurred. To do this, you may need to send a service request to your next level service provider or to IBM service support (*IBMSRV).

Answering problems

Problems reported to an OS/400 service provider may be answered automatically with no operator intervention, or manually by selecting the Answer problem option from the Work with Problem menu.

When you are working with a particular problem record in the problem log and the status of the problem is no longer OPENED, you can select the Answer problem option from the Work with Problem menu to show the Answer Problem menu (if the status of the problem is ANSWERED, this option is shown as Change answer). From this display, you can select an option to search the PTF database on the service provider or work with PTF information for the problem. When the problem is a software problem, the answer may be found in the PTF database on

the service provider. When the problem is a hardware problem, the answer may be a replacement part, and a service representative will need to contact the service requester directly.

Software problems

When a problem involves a program error, you can have the service provider attributes set so that service requests and PTF orders are answered automatically. When you do this and a request is received, the system searches the PTF database and if PTFs are found and released, they are sent to the service requester.

If the PTFs are not already available on the service provider, you need to check whether the necessary PTFs are requested, on hold, or simply do not exist on the system. If the necessary PTFs are requested or on hold, you can use the options available on the Work with PTF Information display or use the PTF commands described in Chapter 6, “Using the program temporary fix (PTF) functions” to work with the PTF status and information.

If you do not want service requests or PTF orders answered automatically, or if the PTFs needed are not found or are on hold, some operator intervention is required. To answer problems manually, do the following:

1. Type the Work with Problem (WRKPRB) command on any command line and press the Enter key. The Work with Problem display is shown.
2. Type an 8 (Work with problem) in the *Opt* column of the problem you want to answer and press the Enter key.
3. Select option 5 (Answer problem) from the Work with Problem menu. The Answer Problem display is shown.
4. Select option 1 (Answer from PTF database). The system searches the PTF database for PTFs with symptom information that matches the symptoms in the problem record.

OR

Select option 2 (Work with PTF information) to create your own list of PTFs that resolve the problem or to add PTF numbers to an existing PTF list.

Notes:

- a. This option is not shown if the problem record is created as a result of a PTF order.
- b. Selecting option 1 when a list of PTFs already exists for a problem allows the system to write over the existing list of PTFs.

When the system searches for PTFs and a match is found, the numbers of the PTFs used to correct the problem are written into the problem record. To view and work with the PTF information, select option 2 (Work with PTF information) from the Answer Problem display.

Once PTFs are found that match the symptoms of a problem, the status of the PTFs are checked. If the necessary PTFs are available for distribution (they exist in save files and are not held for any reason), they can be sent to the service requester using the Send response option on the Work with Problem menu.

Using the Work with PTF Information display

From this display you can work with PTF information associated with the problem you are currently working on. The following operations are available:

- Work with PTFs

- Add PTF information to the problem record
- Remove PTF information from the problem record

The following display shows an example of PTF information for a software problem.

```

                                Work with PTF Information
                                System:  xxxxxxxx
Problem ID . . . . . : 9214237610
Origin . . . . . : RPC.xxxxxxx
Current status . . . . . : ANSWERED
Problem . . . . . : Software problem detected in QUPDRVR.

Type options, press Enter.
  1=Add  4=Remove  8=Work with PTF

Opt  PTF ID   Requested  PTF Save File
  1
    1ZT0000  No          No

                                Bottom

F3=Exit  F5=Refresh  F11=Display product and release  F12=Cancel
F18=Work with PTFs  F22=Send PTFs

```

```

                                Work with PTF Information
                                System:  xxxxxxxx
Problem ID . . . . . : 9214237610
Origin . . . . . : RPC.xxxxxxx
Current status . . . . . : ANSWERED
Problem . . . . . : Software problem detected in QUPDRVR.

Type options, press Enter.
  1=Add  4=Remove  8=Work with PTF

Opt  PTF ID   Product ID  Release
    1ZT0000  PZPR0D1   V1R1M0

                                Bottom

F3=Exit  F5=Refresh  F11=Display status  F12=Cancel  F18=Work with PTFs
F22=Send PTFs

```

The Add PTF Entry display allows you add a Program Temporary Fix (PTF) ID to the list of IDs assigned to the problem specified.

The PTF ID can be specified or *GEN can be used if you want the system to create a PTF identifier for you. The system finds the first available identifier within the range of identifiers for the specified product and release.


```

                                Add PTF Entry
                                System:  xxxxxxxx
Problem ID . . . . . : 9214237610
Origin . . . . . : RPC.xxxxxxx
Current status . . . . : ANSWERED
Problem . . . . . : Software problem detected in QUPDRVR.

Type choices, press Enter.

PTF ID . . . . . : Character value, *GEN
Product ID . . . . . : F4 for list
Release . . . . . : VxRxMx

F3=Exit  F4=Prompt  F12=Cancel

```

The entries in the *Requested* column identify if the PTF has been requested either from another service provider or from IBM service support (*IBMSRV). Either a service request or PTF order has been sent for this problem record.

When the entry in the *Requested* column is Yes, it indicates that these PTFs have been requested from another service provider as a result of this problem record.

When the entry shown in this column is No, there are two possibilities:

The problem was reported by one of your service requesters and was answered either from your PTF database, or by adding PTF numbers using the option to Work with PTF information.

Or

You reported the problem to another service provider, received an answer, and the PTFs associated with this problem record have already been requested or received.

The entries in the *PTF Save File* column indicate whether the PTF is in a save file on your system and whether it can be distributed to your service requesters.

When an entry in the *PTF Save File* column is Yes (Released), the PTF exists on your system in a save file that is released and can be distributed to your service requesters.

When the entry shown in this column is Yes, the PTF exists on your system in a save file that is on hold and cannot be distributed to your service requester (the PTF save file is on hold).

When the entry shown in this column is No, the PTF does not exist in a save file on your system.

The search for PTFs is dependent on keeping the service provider database current. This means periodically installing and distributing the latest cumulative PTF packages, and using the Order Supported Product PTFs (ORDSPTPTF) command to order PTFs for the licensed programs for which you provide support. If you have never had a problem that called for a particular PTF or group of PTFs, then you may need to send a service request or PTF order to your next level service

provider. You may also need to use the Work with Supported Products (WRKSPTPRD) command to add to and update the list of products, product options, and language features for which you provide service. This simplifies system maintenance when you have a large number of systems in a network. For more information about working with PTFs, see Chapter 6, "Using the program temporary fix (PTF) functions" on page 115.

Note: A service request or PTF order answered by IBM service support (*IBMSRV) may result in a larger package of PTFs than the same request answered by an OS/400 service provider. This is because of differences in the methods used for the PTF search and the size of the IBM service support (*IBMSRV) PTF database.

Hardware problems

When you have a service requester with an equipment problem, you need to send a service request to your hardware service provider. In most cases, this is IBM service support (*IBMSRV). The Report problems option on the Work with Problem menu allows you to send all the problem analysis information to IBM service support (*IBMSRV), so that a service representative may be sent to the service requester site with the replacement part.

When you select the option to send a response for a problem related to a piece of equipment or part, the possible causes list is sent to the service requester along with a message indicating that a service representative has been contacted. When the service representative arrives at the service requester site, a new piece of equipment or part will be installed, and the corrected problem verified.

Reporting problems from the service provider

When you have completed problem analysis for a problem and you do not have the PTFs on the service provider to resolve the problem, or the problem is an equipment failure, you need to send a service request to your next level service provider or to IBM service support (*IBMSRV).

You have several alternatives for reporting problems to your next level service provider:

- The Send Service Request (SNDSRVRQS) command
- The Report problem option on the Work with Problem menu
- The function key provided on the possible causes display
- The telephone

The SNDSRVRQS command allows you to send groups of problems from the problem log at the same time. The problems must be in either OPENED or PREPARED status. However, if your next level service provider is IBM service support (*IBMSRV), you cannot send problems with a status of OPENED. A list of service providers is displayed during the preparation of a service request when you specify *SELECT for the remote control point parameter. You can use this command in either a batch or interactive environment.

If you want to send a group of problems to a service provider, you can use the ACTION parameter for the Send Service Request (SNDSRVRQS) command. Specify *PREPARED or *OPENED. When you specify *PREPARED, the system sends all problem records in PREPARED status to the service provider you specified. If you specify *OPENED for the ACTION parameter, you must also specify a service

provider other than IBM service support. The system then sends all problems with OPENED status to the specified service provider.

The Report problem option on the Work with Problem menu and the function key on the possible causes display can only send a service request for the problem you are working on.

Problems reported *from* a service requester should be in either READY or PREPARED status. When you send the service request, you can select the destination of the service request, indicate the severity of the problem, and select the mode of reporting. When you add your own notes to a problem record, these are also sent to the service provider to which you are reporting the problem.

The following display shows the default parameters for the Send Service Request (SNDSRVRQS) command.

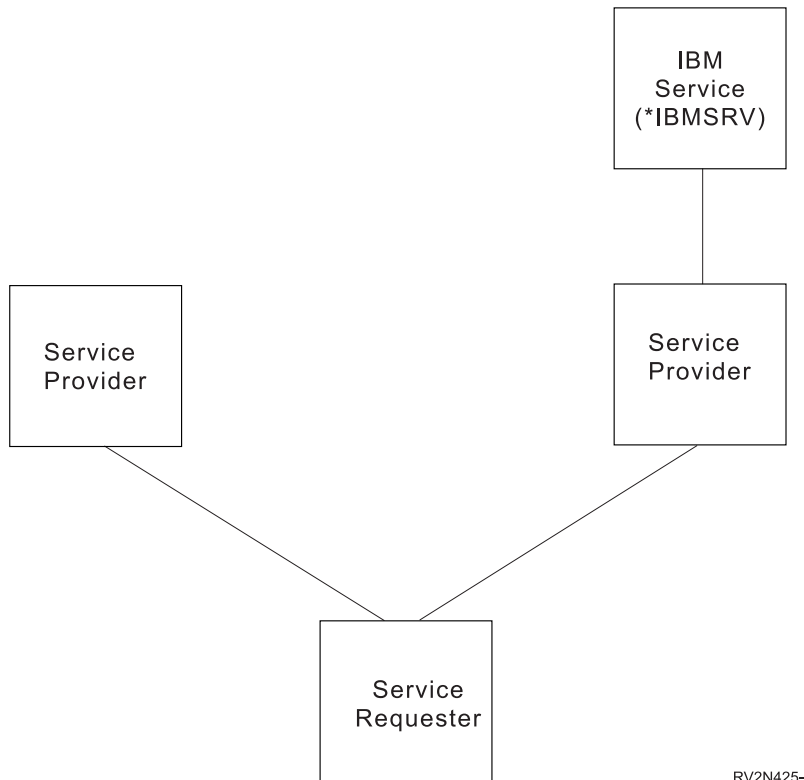
```
Send Service Request (SNDSRVRQS)

Type choices, press Enter.

Action . . . . . *PREPARED, *OPENED, *TEST...
Remote control point . . . . . *IBMSRV      Name, *IBMSRV, *SELECT
Remote network identifier . . . . . *NETATR  Name, *NETATR

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

System Manager allows a software vendor to package an application. Once the application is packaged, the vendor can support the application by using the System Manager service provider and service requester interfaces. Be sure to consider how to properly route problem notifications. In the example network shown in Figure 27 on page 194 it is possible to have more than one service provider.



RV2N425-3

Figure 27. Central Site System and Managed Systems

Problem analysis commands

The following commands are available to start or complement the problem analysis procedures for the iSeries server.

Work with Problem (WRKPRB) command

The Work with Problem (WRKPRB) command allows you to specify a problem or group of problems in the problem log with which you want to work. When a problem record is for a problem occurring on a service requester, you can start remote problem analysis and many other problem handling operations. For more information about this command, see "Using the problem log commands" on page 157

Analyze Problem (ANZPRB) command

The Analyze Problem (ANZPRB) command is typed on any command line and starts problem analysis for user-detected problems. To specify parameters, press F4 (Prompt), or use the menus provided to indicate the service requester on which you want to run remote problem analysis. When the analysis is complete, the results are stored in the problem record and are used to search for PTFs to correct the problem or to prepare a new service request if no solution is found.

Work with Alerts (WRKALR) command

The Work with Alerts (WRKALR) command allows you to display and work with alerts created by your system or received from another system. You can use the Work with Alerts display and the automatic refresh function to show alerts as they

are received on the system. You can then use the option to work with the problem and run remote problem analysis. There are two paths to problem analysis from the Work with Alerts display: select option 9 (Work with problem) to run problem analysis, if available, for a system-detected problem, or press F16 (Analyze new problem) to run problem analysis for a user-detected problem. This command may only be run in an interactive environment.

Verify Communications (VFYCMN) command

The Verify Communications (VFYCMN) command allows an operator to verify communications equipment either locally or remotely. Type VFYCMN on any command line and press F4 (Prompt). The prompt display allows you to select the system on which you want to run remote communications verification.

Depending on the system's configuration, the operator can run the following tests:

- Cable
- Communications input/output adapter
- Communications interface trace
- Link
- Local modem
- Remote modem
- Link Problem Determination Aid-2 (LPDA-2)

Verify Tape (VFYTAP) command

The Verify Tape (VFYTAP) command starts procedures that check the specified tape unit and verify if the tape unit is operating. Displays are presented and the operator is prompted for the resource name, which appears when *RSRCNAME is typed in the device name position and the Enter key is pressed. This gives you the ability to enter either a device description name or the resource name. This command is for local tape verification and is presented here as a complement to the problem analysis procedures. You can have an operator at a service requester use this command to verify a tape unit problem has been resolved.

Problem management APIs

The following APIs are available to manage problems on the iSeries server:

QPDLOGER

Creates a problem record, builds a symptom string, and collects data pertaining to the problem.

QpdReportSoftwareError

Performs the same function as QPDLOGER but is more flexible and is designed to be used by ILE programs.

QsxStartProblemLogServices

Sets up the environment so the system can create, change, delete, and retrieve problem records.

QsxEndProblemLogServices

Ends an instance of the problem log services.

QsxAddProblemLogEntry

Provides a means of adding information to an existing problem record.

QsxCreateProblemLogEntry

Adds a new problem record to the problem log.

QsxChangeProblemLogEntry

Updates an existing problem record.

QsxDeleteProblemLogEntry

Deletes a problem record.

QsxRetrieveProblemLogEntry

Extracts data from a specified problem record.

Note: The QPDLOGGER and QpdReportSoftwareError APIs do not generate a problem record. However, they still generate a symptom string and collect information. This information is different depending whether or not the system is running threads.

Chapter 10. Controlling operations centrally

With System Manager, you can control operations on all systems in your network from the central site system. Managed System Services must be installed on the central site system and on all other managed or remote systems in the network. Also, SNA distribution services (SNADS) must be used to allow communication with the remote systems.

You can use System Manager to plan daily business tasks or activities and then put the plan into the system.

The plan enables you to identify the operations that are to be performed, the systems on which they are to be performed, and when they are to be performed. Once you have your plan designed, you let the system run the plan for you. The operations are tracked so you know what is happening. Messages are returned from remote systems if any problems are encountered.

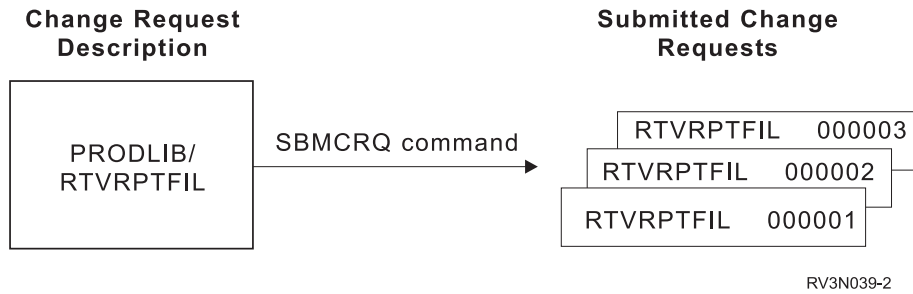
The plan is referred to as a change request description. A **change request description** consists of a series of steps that describe a change to be made to the network. A change could be a new software application, fixes to an existing application, new file data, or any related information. You can use the change request description to:

- Run commands
- Distribute objects
- Send, apply, or remove PTFs
- Install software applications
- Install objects
- Uninstall objects
- Send license keys

An **installable object** is an object to be installed that is associated with the name of the target library, folder, or path where it must be created when the installable object is installed. The change request description, which is maintained only at the central site iSeries server, consists of a list of activities that describe the steps needed to complete the change or operation. Some examples of activities are:

- Sending a file
- Retrieving data
- Running a program
- Installing a product
- Removing a PTF
- Installing objects
- Uninstalling objects
- Sending license keys

A change request must be submitted in order to run the activities. A submitted change request is an instance of a change request description that runs. A change request can be submitted multiple times. A sequence number uniquely identifies each submitted change request.



After a change request is submitted, you can monitor the progress. If problems occur, you can view messages describing the problem and recovery actions. To summarize, when you use change request descriptions to control operations, you follow these steps:

1. Create a change request description
2. Add the activities to that change request description
3. Submit the change request
4. Track the submitted change request

If you do not need to do any complex planning and you want to get started quickly, see “Accessing fast path distribution commands” on page 219 for information about the fast path commands.

Creating a change request description

To work with change descriptions, type `go svsm` from any command line and select option 6 (Change request administration) from the System Manager main menu.

The following display is shown, where you can monitor or work with your change requests:

```

CRQADM                Change Request Administration

                                System: XXXXXXXX

Select one of the following:

  1. Work with change request description
  2. Work with distribution catalog entries
  3. Configure System Manager/400

 10. Work with submitted change requests
 11. Work with distribution queues

Selection or command
=====> _____

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=User support
F16=AS/400 main menu

```

When you select option 1 (Work with change request description) and press the Enter key, the Work with Change Request Descriptions command is started.

When you press the Enter key again, the following display is shown:


```

Work with CRQ Descriptions
System: XXXXXXXX

Type options, press Enter.
1=Create 2=Change 3=Copy 4=Delete 6=Print 8=Work with activities
12=Work with submitted CRQs 13=Schedule CRQ submission 14=Submit ...

Change
Request
Opt Description Library Text
1_
--- INSAPP PRODLIB Send and install new application
--- RUNBACKUP PRODLIB Run data backup programs
--- STOREUPD PRODLIB Daily price file updates for stores

Parameters or command
===>
F3=Exit F4=Prompt F6=Print list F11=Display user and problem ID
F12=Cancel F17=Position to F23=More options F24=More keys
Bottom

```

You can use option 1 (Create) on the Work with CRQ Descriptions display to create a new change request description. When you select option 1 and press the Enter key, the Create CRQ Description display is shown.

For this example, the change request description is APUPDT. It is an application update that is sent to several stores located in different cities throughout the area.

```

Create CRQ Description (CRTCRQD)

Type choices, press Enter.

Change request description . . . . . APUPDT Name
Library . . . . . QGPL Name, *CURLIB
User profile . . . . . *SBM *SBM, *OWNER
Problem identifier . . . . . *NONE Character value, *NONE
Problem origin:
Network identifier . . . . . *NETATR Name, *NETATR
Control point name . . . . . *NETATR Name, *NETATR

Additional Parameters

Authority . . . . . *EXCLUDE Name, *EXCLUDE, *LIBCRTAUT...

Text description . . . . . Application update

Parameters or command
===>
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
F13=How to use this display F24=More keys
Bottom

```

Adding activities to change request descriptions

After creating the change request description, the Work with CRQ Description Activities display is shown to allow you to add activities to the change request description. Use option 1 to add activities.

An **activity** is an operation initiated by a central site system. An activity is a single step within a change request description. Examples of activities are:

- Sending an object
- Deleting a file or a remote file
- Installing an application
- Doing an IPL of the system
- Running a program
- Sending a program
- Applying a PTF
- Sending a license key
- Installing an installable object
- Uninstalling an installable object

If a type is not specified on the Work with CRQ Description Activities display then a selection list is shown. You can choose a type from the list. For this example, use *OBJ for the value in the *Type* field.

You can also add activities after the change request description is created by selecting option 8 (Work with activities) from the Work with CRQ Descriptions display.

```

Work with CRQ Description Activities
System: XXXXXXXX

Change request description . . . . .: APUPDT
Library . . . . .: QGPL
Text . . . . .: Application update

Type options, press Enter.
1=Add 2=Change 3=Copy 4=Remove 5=Display details 8=Display nodes

Opt  Activity  Type  Text
1    *GEN     *OBJ  Application update

Bottom

Parameters or command
====>
F3=Exit  F4=Prompt  F5=Refresh  F6=Print list  F9=Retrieve
F11=Display conditions  F12=Cancel  F17=Position to  F14=More keys

```

When you select optional 1 to add an activity and specify *OBJ in the *Type* field, the following display is shown:

```

Add Object CRQ Activity (ADDOBJCRQA)

Type choices, press Enter.

Change request description . . . . . > APUPDT  Name
Library . . . . . > QGPL      Name, *LIBL, *CURLIB
Activity . . . . . > *GEN     Name, *GEN, *LAST
Action . . . . . *SND      *SND, *RTV, *DLT, *RUN ...
Object . . . . .          Name, *GLOBAL, *COMPNAME
Library . . . . .          Name, *LIBL, *CURLIB

Bottom

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

When adding an activity, you can specify an activity name or allow the system to generate the name for you. One advantage of allowing the system to generate the name is that they are created in alphabetical order. The activities are run sequentially if you also use the default condition described in “Conditioning” on page 202.

As you specify values in the entry fields and press enter, prompts for additional entry fields are displayed. The fields that are displayed depend on the values you specified in the preceding fields.

When adding activities to your change request description, you should consider the following:

- Scheduling activities during off hours
- Activity conditions
- Node list or node names
- Standard names versus global names
- Text description

Scheduling activities

You use the following values to schedule the activities:

- Start times
- Local and remote time zone values

Start times

When you add an activity to a change request description, you can select values that allow the activity to run at the time you specify. There are start times for the central site system and start times for the managed systems in the network. You can specify a window of time in which the activity must occur (that is, it must start after a certain time and before a certain time). If an activity does not start running in the specified time window (for example, it is held), the status is changed to ended and the end code is set to 99.

Local and remote time zone values

Use the *LCLSYS value for the remote start time parameter to specify the start time in the time zone of the central site system. This is useful for having activities start at the same instant throughout the network even though the systems are in different time zones.

The Coordinated Universal Time Offset (QUTCOFFSET) system value must be set correctly on all systems in the network for the *LCLSYS value to work properly.

Use the *MGDSYS value for the remote start time parameter to specify the start time in the time zone of the managed system. This is useful when you want the activity to run at a time that is appropriate for the managed system based on its time zone.

Conditioning

An **activity condition** (or a condition) defines criteria that must be met before the activity can start running. An activity can be conditioned on the results of other activities. The default condition is that the previous activity (alphabetically) in the change request description must end successfully. The order of the activities in the list may be different from the order in which they are run if the default conditions are not used.

All of the conditions that you specify for an activity must be met before the activity can start. Up to five conditions can be specified. The special values for the end codes are *SUCCESS, *FAIL, *NOTRUN. For example, if you specify that activity 01 must have an end code of *SUCCESS, then activity 01 must complete successfully before activity 02 can start (that is, the end code for activity 01 must be in the range of 0 through 9.)

You can have an activity wait until a previous activity has completed on all systems by specifying the *ALLNODES value or on a specific system by specifying the *SAMENODE value for the conditioning mode parameter.

The *ALLNODES value indicates that the conditioning activity specified must complete on all nodes before this activity starts. For example, you may want to apply a PTF after it is received on all systems.

The *SAMENODE value indicates that when the conditioning activity specified completes for a given node, the activity specified on the ACTIVITY parameter may run for that same node even though the conditioning activity specified has not completed for all other nodes. For example, you may need to run an area sales report as soon as the daily report files are retrieved from specific departments.

Node list or node names

Each activity added to a change request description must define the system or systems for which it is intended. You can either specify a node list object name or a list of one or more systems identified by a control point name and a network ID. A node list object contains a list of systems. Only node list entries with an address type of *SNA are used.

You can use the node-list-name (NODL) parameter to specify the name of a node list object containing the list of systems on which the activity is to be performed. Use a node list when you send requests to a list of systems (similar to a distribution list when you send a note).

To create a node list, use the Create Node List (CRTNODL) command and identify the library you want to use to store the list. You can also describe the list in the text description field. When you use this command, an object is created and is used to store the node names that identify a set of systems in your network.

-or-

You can use the CPNAME parameter to specify the APPN control point names of the managed systems on which the activity is to be performed. Control point names cannot be specified if a node list is specified.

Standard names versus global names

When performing object distribution functions, you have a choice of naming methodologies. The naming methodology used determines which functions are available. The choices are:

Standard

Standard naming is the way objects are commonly identified on the iSeries server. A library, object name, and object type uniquely identify the object. In the case of files, a member name is also required.

An example of a standard object name is:

```
OBJ(LIBRARY1/OBJECT2) OBJTYPE(*PGM)
```

Global

Global names consist of a series of up to 10 tokens that form a hierarchical name. These names are independent of the system with a length up to 64 characters, enabling them to uniquely identify an object within a network.

The names are similar to the directory and subdirectory names that you see on various operating systems.

An example of a global name is:

```
GLBNAME(COMPANY1 DIVISION2 DEPT3 REPORT4  
SALES PART Y1994M10D16)
```

Both naming methods can be used on the same system. An object referred to by a catalog entry can be retrieved using the global name or the OS/400 standard name. Objects that are stored in the repository can be referred to only by their global name.

Because global names are always used between the central site system and the managed system, OS/400 standard names are implemented internally using global names. A special set of tokens at the beginning of a global name indicates that the global name contains a standard OS/400 object name. You determine the prefix tokens so they do not interfere with other globally named objects in the network.

Tip

Prefix tokens are used only for global names created by the system. As a result, the prefix tokens cannot be used when defining a global name with the Add Distribution Catalog Entry (ADDDSTCLGE) command.

You should not change the prefix tokens after entries have been added to the distribution catalog.

For more information about name values, see the *Managed System Services for AS/400 Use* book.

Text Description

A text description is a 50-character description of the activity. Be specific in describing the activity so that you recognize it when you display the status.

Because the system identifies all change request activities by number, the text description is helpful when you are tracking the activities. If you specify *GEN, the text is generated for you based on the activity type.

Defining your own activity types

A change request description consists of one or more activities. Each activity has a type. The activity type specifies the type of operation to be performed by the activity. Currently, the following activity types are available:

*CHGMGT

Change management generic activity type

Notes:

1. The *CHGMGT activity type is useful for performing change management activities to non-iSeries platforms using the NetView/DM support.
2. The QNSADDCM API or the Add Change Request Activity (ADDCRQA) command can be used to add this type of activity.

*CMD Run a command

*OBJ Object distribution

*PRD Product distribution

*PTF PTF distribution

*RSC Resource activation

*LICKKEY

License key distribution

You can add additional activity types by registering exit programs that handle the processing required for the new activity type.

Activities can be added, changed, displayed, or run. An exit program is needed to perform each of the functions. Another way to add activities is by using the Add Change Request Activity (ADDCRQA) command.

Four exit points are defined by IBM to which you can add exit programs. Each exit point corresponds to the function to be performed by the exit program. One exit program can be added for each new activity type. The exit points are as follows:

Table 5. Exit Points

Exit point name	Format name	Description
QIBM_QNS_CRADDACT	ADDA0100	Add an activity
QIBM_QNS_CRCHGACT	CHGA0100	Change an activity
QIBM_QNS_CRDSPACT	DSPA0100	Display an activity
QIBM_QNS_CREXCACT	EXCA0100	Run an activity

The Add Exit Program (QUSADDEP) API is used to add an exit program to an exit point. Another way to add an exit program is to use the Work with Registration Info (WRKREGINF) command and choose option 8 (Work with exit programs).

Exit program data

When adding an exit program, program data must be specified. The program data is used to:

- Contain the activity type name the exit program or command is associated with. This is the name shown on the Select Activity Type display.
- Contain the message ID, message file, and library for the activity type description. The message text is also shown on the Select Activity Type display.
- Contain the exit type, either a program or command.

Note: Exit points QIBM_NS_CRDSPACT and QIBM_NS_CREXCACT do not support an exit type of command.

The program data must have the following format:

```
CHAR(10) Activity type
CHAR(27) Activity type description
CHAR(7)  Msg ID
CHAR(10) Msg file name
CHAR(10) Msg file library
CHAR(1)  Exit type (1=program, 2=command)
```

How exit programs are called

The interface used to call an exit program is defined by IBM. See Appendix E, “System Manager APIs and exit programs” on page 291 for more information about each exit program and a description of the parameters each exit program must support. The following exit programs are defined:

- Add Activity Exit Program
- Change Activity Exit Program
- Display Activity Exit Program
- Run Activity Exit Program

Displaying CRQ description activity details

After adding an activity, you can use option 5 from the Work with CRQ Description Activities display to show the Display CRQ Description Activity Details display.

```
Display CRQ Description Activity Details
System: XXXXXXXX

Change request description . . . : APUPDT
Library . . . . . : QGPL
Activity name . . . . . : QACT000010
Activity text . . . . . : Distribute MYPROG source to all systems

Activity type . . . . . : *OBJ
Node list . . . . . : USNODES
Library . . . . . : NODELIBR
Scheduled start:
Start after date and time . . : 10/15/94 22:15:00
Start before date and time . . : 10/16/94 06:15:00
Hold . . . . . : *NO

Bottom

Press Enter to continue.

F3=Exit      F9=Display conditions    F10=Display nodes
F11=Display type specific data    F12=Cancel
```

The *Node list* field specifies the OS/400 node list object that defines the destination nodes for the activity.

If a node list object was not specified when the activity was added, the *Node list* field is replaced by the *Node* field. The *Node* field contains the name of the system that is the target of the activity (specified as a network ID and control point name).

Displaying nodes

To display the list of nodes that the activity acts on, press F10 from the Display CRQ Description Activity Details display.

```

                                Display Nodes
                                System:  XXXXXXXX

Change request description . . . : APUPDT
Library . . . . . : QGPL
Activity name . . . . . : QACT000010
Activity text . . . . . : Distribute MYPROG source to all systems

Control   Network   Control   Network   Control   Network
point    ID          point    ID          point    ID
STORE010 STORENET  STORE111 STORENET
STORE031 STORENET  STORE112 STORENET
STORE099 STORENET  STORE113 STORENET
STORE100 STORENET  STORE114 STORENET
STORE101 STORENET
STORE102 STORENET
STORE103 STORENET
STORE104 STORENET

                                Bottom

Press Enter to continue.

F3=Exit  F12=Cancel  F17=Top   F18=Bottom

```

Displaying conditions

To determine the conditions for an activity, press F9 from the Display CRQ Description Activity Details display. The Display Conditions display is shown.

```

                                Display Conditions
                                System:  XXXXXXXX

Change request description . . . : APUPDT
Library . . . . . : QGPL
Activity name . . . . . : QACT000040
Activity text . . . . . : Distribute MYPROG source to all systems

Conditioning
Activity      Relation  Code      Condition
QACT000010   *EQ      *SUCCESS  *SAMENODE
QACT000020   *EQ      00        *SAMENODE
*PRV         *EQ      *ANY      *ALLNODES

                                Bottom

Press Enter to continue.

F3=Exit  F10=Display job  F12=Cancel

```


Each condition specified must be met before the activity listed in the *Activity name* field can run. Each condition identifies a conditioning activity, which must end in a manner that makes the relation true to allow the current activity to run.

The *Activity* column specifies the name of the conditioning activity.

The *Relation* column specifies the relational operator in the activity condition. It defines the relationship between the end code of the conditioning activity and the end code specified in the condition.

The *Code* column specifies the end code required for the conditioning activity.

The *Condition Mode* column specifies where the condition must be met before an activity can be performed. The condition mode is used to determine when the activity can start running for a particular node.

The **ALLNODES* value indicates that the conditioning activity specified must complete on all nodes before this activity starts. For example, you may want to apply a PTF after it is received on all systems.

The **SAMENODE* value indicates that when the conditioning activity specified completes for a given node, the activity specified on the *ACTIVITY* parameter may run for that same node even though the conditioning activity specified has not completed for all other nodes.

End code definitions

The condition code is the value that is compared to the actual end code of the conditioning activity.

- **SUCCESS* means the activity ended successfully
This end code can only be specified with relational operator **EQ* or **NE*.
- **FAIL* means the activity failed
This end code can only be specified with relational operator **EQ* or **NE*.
- **NOTRUN* means the activity never started
This end code is only specified with relational operator **EQ* or **NE*.
- **ANY* means the activity ended with any end code.
This end code is only specified with relational operator **EQ*.
- End-code (an integer value from 0 through 99) indicates the result of an activity (success or failure). The end code ranges and descriptions are:

00 Activity completed successfully.

01-09 Activity completed with warning messages.

10-29 Activity did not complete successfully.

30-39 Activity was canceled by a user before it completed.

- 30 = Activity ended with **CNTRLD* option
- 35 = Activity ended with **IMMED* option
- 39 = Activity ended with **FRCFAIL* option

40-49 Activity was not run due to errors detected by the application.

- 40 = Not run for security reasons

90-99 Activity was not run because conditions or schedules were not met.

- 95 = Scheduled start time expired

- 99 = Conditions cannot be met

Note: The system does not log a message when the request has an end code of 99. For all other non-zero end codes, messages are logged and can be viewed by selecting option 10 (Display messages).

Displaying type specific data

To display information specific to a change request description activity (for example, an object distribution), press F11 from the Display CRQ Description Activity Details display. In the following example, the OS/400 object name MYPROG is used in an activity that sends the object to multiple remote systems.

```
Display Type Specific Data

Change request description . . . : APUPDT
Library . . . . . : QGPL
Activity name . . . . . : ACT01
Activity text . . . . . : Distribute MYPROG source to all systems

Action . . . . . : Send object

Object . . . . . : MYPROG
Library . . . . . : MYLIB
Object Type . . . . . : *FILE
Target release . . . . . : *CURRENT
Data object class . . . . . : AS/400 object

Replace . . . . . : *NO

Press Enter to continue
F3=Exit      F12=Cancel
```

Submitting change requests

The change request description defines your plan. In order to run the plan you need to submit the change request. You can use the Submit Change Request (SBMCRQ) command or you can use option 14 on the Work with CRQ Descriptions display.

The System Manager program and the Managed System Services program must be started in order for the change request to run. If the Managed System Services program is not started, the activities in the change request remain in READY status. If the System Manager program is not started, the activities in the change request remain in SCHEDULED status.

If there are activities in the change request description that perform object distribution functions, then the QSNADS subsystem must also be started. Activities will remain in RUNNING status indefinitely if the QSNADS subsystem is not started.

What happens if the request does not run

Here are some examples of situations that may occur:

- If System Manager is not started, a diagnostic message is issued informing the user that it must be started before a submitted change request can run.
- If, after a change request has been submitted, the status of the activities remains at Ready, this may indicate that Managed System Services is not started. Run the Start Managed System Services (STRMGDSYS) command to use the Managed System Services functions.
- If, after a change request has been submitted, the status of the activities remains at Running, this may indicate that the QSNADS subsystem is not active. Use the Start Subsystem (STRSBS) command to start the QSNADS subsystem. The Running status may also indicate that there is a communications problem. Check your APPC controllers and devices to ensure they are active.
- Check the job log for job QCQSFSRV with the user profile name that the activity is running under. This job log may also contain messages to help you determine why a change request does not run.
- If the change request description has been submitted and the status of the activities remains at Ready, it may indicate that the maximum active jobs for this job queue entries is exceeded. To identify the problem, use the Work with Job Queue (WRKJOBQ) command to view the jobs column for QNMSVQ queue entry has a number greater than zero. The problem could be that too many service jobs are opened at the same time.

The alternatives to fix this problem:

- Change the maximum active jobs for this job queue entry. Use the Change Job Queue Entry (CHGJOBQE) command specifying the QSYSWRK subsystem and QNMSVQ job queue entry and change the value in the Maximum active jobs (MAXACT) parameter to a number greater than the existing number.
- or–
- End some of the QCQSFSRV jobs in the QSYSWRK subsystem that are not used at this time.

Activity considerations

Note: The more activities and nodes associated with a change request description, the bigger the challenge is to manage it. Also, the submit process takes longer for large and complex change requests.

To help ensure change request descriptions are manageable and the time to submit is reasonable, some limits have been defined pertaining to the size of a change request description.

- The maximum number of activities that you can add to a change request description is 2,500 activities.
- The maximum number of nodes that can be specified in a node list is 500.
- The maximum number of destination nodes in the entire change request description is 30 000.

If any of these limitations are exceeded, the user is notified when the change request is submitted.

Activities are not guaranteed to run in the same jobs. Therefore, references to objects in library QTEMP should be avoided since the scope of QTEMP is only within the job.

Tracking submitted change requests

Extensive tracking capabilities are available after you have submitted a change request. Use these steps to access these capabilities:

1. Select option 6 (Change request administration) from the System Manager menu.
2. Select option 10 (Work with submitted change requests) to search or select specific change requests.

Or, you can use the Display Submitted Change Requests (DSPSBMCRQ) command to show the Display Submitted Change Requests display. This display shows the status of the change requests you have submitted.

Note: In addition to the extensive tracking and status information available from the commands and displays, you can access the submitted change request files directly. This allows you to add advanced reporting capabilities.

The following files are defined with public authority equal to *USE. The files reside in library QSMU.

File Name	Description
QANSCRL	Change request information
QANSCRAL	Activity information
QANSCRNL	Node information
QANSCRCL	Conditioning information
QANSCRML	Message information

Working with submitted change requests

If you select option 10 (Work with submitted change requests) from the Change Request Administration menu and press the Enter key twice, the following display is shown.

```

Work with Submitted Change Requests
System: XXXXXXXX
Type options, press Enter.
3=Hold 4=Delete 5= Display details 6=Release 8=Work with activities
10=Display messages 13=End ...

Change
Opt Request      Number  Status  Highest End      Last
---  Request      Number  Status  End      End
---  APUPDT        000010  Ended   20       20
---  APUPDT        000020  Active
---  FILEUPD       000010  Active
---  RUNRPT        000010  Active

Parameters or command
====>

F3=Exit   F4=Prompt   F5=Refresh  F6=Print list  F11=Display user
F12=Cancel F17=Position to F23=More options F24=More keys
Bottom

```

From this display, you can select options to:

- Display details about the change request and problems associated with the change request.
- Work with activities of the change request.
- Display messages logged regarding the change request.
- Hold, release, or end activities.
- Delete a change request (only if it is ended).

Displaying submitted CRQ details

If you select option 5 (Display details) for a specific change request, the following display is shown:

```

Display Submitted CRQ Details
System: XXXXXXXX
Change request . . . . . : APUPDT
Number . . . . . : 000010

Status . . . . . : Ended
Highest end code . . . . . : 00
Last end code . . . . . : 00
Change request description . . . : APUPDT
Library . . . . . : QGPL
User profile . . . . . : *OWNER
Submitter . . . . . : SAMUEL
Submitted date and time . . . . : 10/08/94 08:00:00
Problem ID . . . . . : 1234567890
Origin . . . . . : STORENET.STORE004
Text . . . . . : Application update

Press Enter to continue.

F3=Exit   F12=Cancel   F18=Display activities  F19=Display problem
Bottom

```

Working with submitted CRQ activities

From the Work with Submitted Change Requests display, select option 8 (Work with activities) to work with submitted change request activities. The following display is shown:

```

Work with Submitted CRQ Activities
System: XXXXXXXX
Change request . . . . . : APUPDT
Number . . . . . : 000010
Text . . . . . : Application update

Type options, press Enter.
3=Hold 5=Display details 6=Release 8=Work with nodes for activity
10=Display messages 13=End

Activity
Opt Name Type Node Status Highest
End Code
--- QACT000010 *OBJ STORENET.STORE... Ready
--- QACT000020 *PRD STORENET.STORE... Scheduled
--- QACT000030 *CMD STORENET.STORE... Ended 20
--- QACT000040 *OBJ STORENET.STORE... Ended 00
--- QACT000050 *CMD STORENET.STORE... Ended 00

Parameters or command
====>
F3=Exit F4=Prompt F5=Refresh F6=Print list F9=Retrieve
F11=Display conditions F12=Cancel F17=Position to F24=More keys

```

The status of an activity is based on the status of the activity for each node. The activity has the same status as the status of the node with the highest weighted value. The activity status values that are listed below show the highest weighted value first.

- **Running**

The node currently is running on the local system. SNADS is attempting to deliver the entry to the corresponding system. If the activity remains in running status for a long time, you may have a configuration error or problem with SNADS.

First, check the distribution queue at the remote site to verify that there is a value of *SVDS which points back to the central site. Second, check to make sure that the QMSF jobs are running under QSYSWRK subsystem. Third, make sure that the SUBSYSTEM QSNADS is running.

Note: If this is a distribution from a system that is acting as central site and CCServer at the same time, and this distribution is for a system that acts as a CC Client, the distribution changes from RUNNING to ENDED. You will not view the SENT status. If the systems are in this scenario and the activities remain in RUNNING status for a long time, then ensure that the CC Client is in AVAILABLE status.

- **Not sent**

The activity was not sent to the destination node because an SNA distribution services (SNADS) queue that is along route is in the Retry/fail state. Typically, you receive this status as a result of a communication error.

- **Sent**

The system has sent the activity to the destination node.

- **Started**

Processing of the activity has started at the central site system.

- **Ready**

The node is ready to start. All conditions and needed schedules are defined.

- **Wait**

The node is waiting for you to specify one or more conditions. Until you specify the conditions that are needed, the activity does not run.

- **Scheduled**

The activity for the node begins at the time you have specified. Sometimes you may receive a status of SCHEDULED when you have one or more conditions that still must be specified.

- **Not started**

The node was not processed by the application because there is an error. You can find further information about the error in the Change Request log. Follow the steps below to display the error message.

1. To display the change request activity message, type the Display Submitted Change Messages (DSPSBMCRQA) command.
2. To display the details select option 5 (Display details) next to the proper activity.
3. After correcting the problem that caused the error, you must release the node. To release the node, type the Release Submitted Change Request Activity (RLSSBMCRQA) command.

- **Held**

The system displays this status when the user holds the node.

- **Ending**

The system displays this status when the user tries to end the activity for a node that was previously in Started or Running status. The end is in progress.

- **Ended**

The node that was running has ended.

For example, an activity has the following nodes as the destination and they each have the specified status.

Table 6. Status for a destination node

Nodes	Status
STORENET.STORE01	Ended
STORENET.STORE02	Ready
STORENET.STORE03	Scheduled
STORENET.STORE04	Held

Since the Ready value has a higher weighted value than the other status values, the activity has a ready status.

For details about each field on the display, refer to the online help information.

Note: If the destination node is the local system and the activity type is *CMD, the value *LCL is displayed in the Node column for that activity.

Working with nodes for activity: From the Work with Submitted CRQ Activities display, you can select option 8 (Work with nodes for activity) to see the following display:

```

Work with Nodes for Activity
System: XXXXXXXX
Change request . . . . . : APUPDT
Number . . . . . : 000010
Activity name . . . . . : QACT000010
Activity text . . . . . : Application update

Type options, press Enter.
3=Hold 6=Release 8=Display conditions for node 10=Display messages
13=End ...

Opt Control Network Status End
point ID Code

STORE010 STORENET Ready
STORE031 STORENET Ready
STORE099 STORENET Ended 00
STORE100 STORENET Ended 00

More ...

Parameters or command
====>
F3=Exit F4=Prompt F5=Refresh F6=Print list F9=Retrieve
F11=Display job F12=Cancel F23=More options F24=More keys

```

This display allows you to see the status of each node. If the activity ended for a node, you can also see the end code, which indicates if the activity ended successfully. For nodes that end with an error (that is, not 00), you can use option 10 to display the error messages. See “Working with submitted CRQ messages” on page 215 for more information about messages.

See “End code definitions” on page 207 for the meanings of the possible values for the end code.

It is important to look at the end code for all activities with status ended. Although all end codes other than 00 require some kind of problem determination, end code 20 is the most common error condition. See “Working with submitted CRQ messages” on page 215 for more information about displaying messages.

Note: The system does not log a message when the request has an end code of 99. For all other non-zero end codes, messages are logged and can be viewed by selecting option 10 (Display messages).

When you press F11 (Display job), job status information is shown. A *job* is the name of the job at the central site that is currently running the activity. This field may be blank for activities that have been started, which means that a request has been sent to a managed system to process the activity.

When you press F11 (Display dates), actual start and end times are displayed.


```

Work with Nodes for Activity
System: XXXXXXXX
Change request . . . . . : APUPDT
Number . . . . . : 000010
Activity name . . . . . : QACT000010
Activity text . . . . . : Application update

Type options, press Enter.
3=Hold 6=Release 8=Display conditions for node 10=Display Messages
13=End ...

Control Network
Opt Point ID ---- Start ----- End -----
Date Time Date Time
-- STORE010 STORENET
-- STORE031 STORENET
-- STORE099 STORENET 10/15/94 23:15:35 10/16/94 01:04:20

More...

F3=Exit F4=Prompt F5=Refresh F6=Print list F9=Retrieve
F11=Display status F12=Cancel F23=More options F24=More keys

```

This view of the Work with Nodes for Activity display allows you to see when the activity started and ended for each node.

Working with submitted CRQ messages

When you select option 10 (Display messages), the Display Submitted CRQ Messages display is shown.

```

Display Submitted CRQ Messages
System: XXXXXXXX
Change request . . . . . : APUPDT
Number . . . . . : 000001
Activity name . . . . . : QACT000010
Managed system node name:
Control point name . . . . . : STORE099
Network identifier . . . . . : STORENET

Type option, press Enter.
5=Display details 6=Print

Opt Message
-- Request not successful
_5 Object already exists

Bottom

F3=Exit F5=Refresh F6=Print list F9=Command F12=Cancel F17=Top
F18=Bottom

```

The messages are displayed in order by date and time logged. New messages are displayed at the bottom. The applications registered to System Manager have the ability to log messages that are shown on this display.

If messages are for all nodes, *ALL is displayed for the *Control point name* field and the *Network identifier* field. If messages are for all activities, *ALL is displayed for the *Activity name* field.

Displaying submitted CRQ message details: From the Display Submitted CRQ Messages display, you can select option 5 (Display details) to see the following display:

```

                                Display Submitted CRQ Message Details
                                System:  XXXXXXXX
Change request . . . . . :  APUPDT
Number . . . . . :  000001
Activity name . . . . . :  QACT000010
Managed system node name:
  Network identifier . . . . . :  STORENET
  Control point name . . . . . :  STORE099

Message ID . . . . . :  SMU16CF
Severity . . . . . :  00
Date/time sent . . . . . :  10/16/94  01:03:50
Message . . . . . :  Object already exists.

From job . . . . . :  QCQSVSRV
  User . . . . . :  SAMUEL
  Number . . . . . :  003252

                                Bottom

Press Enter to continue.

F3=Exit  F10=Second level text  F12=Cancel

```

To view additional message information, press F10 (Second level text). Second level text gives you information about what to do to correct the problem. In this example (object already exists), the additional message information explains that you attempted to create an object that already exists but you did not request to replace the existing object.

1. When you use the Work with Submitted Change Requests (WRKSBMCRQ) command and select option 8 (Work with activities) on the change request, a display similar to the following is shown:

```

                                Work with Submitted CRQ Activities
                                System:  NORA
Change request . . . . . :  NOTFOUND
Number . . . . . :  000070
Text . . . . . :

Type options, press Enter.
  3=Hold  5=Display details  6=Release  8=Work with nodes for activity
  10=Display messages      13=End

Activity
Opt  Name      Type      Node      Status      Highest
     QACT000010 *OBJ     APPN.NORA Ended        End
                                           Code
                                           20

                                Bottom

Parameters or command
====>
F3=Exit  F4=Prompt  F5=Refresh  F6=Print list  F9=Retrieve
F11=Display text  F12=Cancel  F17=Position to

====>

```

Type option 10 over the activity and see the message list.

```

                                Display Submitted CRQ Messages
                                System:  NORA
Change request . . . . . :  NOTFOUND
Number . . . . . :  000070
Activity name . . . . . :  QACT000010
Managed system node name:
Control point name . . . . . :  *ALL
Network identifier . . . . . :  *ALL

Type options, press Enter.
5=Display details  6=Print

Opt  Message
    Object not found.
    Additional information received from APPN.NORA.
    Additional information received from APPN.NORA.

                                Bottom
F3=Exit  F5=Refresh  F6=Print list  F9=Command  F12=Cancel  F17=Top
F18=Bottom

```

2. If you type option 5 on each message and then press Enter, you will see a description for each message:

```

                                Additional Message Information
Message ID . . . . . :  SMU16C1      Severity . . . . . :  40
Date Sent . . . . . :  12/09/94     Time Sent . . . . . :  12:39:55

Message . . . . . :  Object not found.

Cause . . . . . :  The object specified on the request was not found.  If you
used a standard object name in the request, it is possible that the prefix
tokens on the central site system and the managed system do not match.
Recovery . . . . . :  Do one of the following:
-- If you used a standard object name in your request use the Display
Managed System Attributes (DSPMGDSYSA) command on both systems to check the
prefix tokens and if they do not match, use the Change Managed System
Attributes (CHGMGDSYSA) command to change them so they do match.
-- Use the Work with Change Request Descriptions (WRKCRQD) command to
change the activity.  Specify an object name or a global name that
identifies an existing object.
Then use the Submit Change Request (SBMCRQ) command to submit the activity

Press Enter to continue.

F3=Exit  F10=Display job  F12=Cancel

```

```

Additional Message Information
Message ID . . . . . : SMU16F1      Severity . . . . . : 00
Date Sent . . . . . : 12/09/94     Time Sent . . . . . : 12:39:53

Message . . . . . : Additional information received from APPN.NORA.

Cause . . . . . : Additional information was received because the activity
                  failed or the activity completed with warnings. The following information
                  was received from control point NORA with network ID APPN: MSS011B
                  Distribution catalog entry not found.Cause . . . . . : Distribution
                  catalog entry with global name QXXZZIN01 OBJECT NOTFOUND REF 1 was not found.
Recovery . . . . . : Do one of the following, then try the request again: --
                  Use the Add Distri
Recovery . . . . . : See the additional information received. Correct the
                  error, then try the request again.

                                                                    Bottom

Press Enter to continue.

F3=Exit  F10=Display job  F12=Cancel

```

```

Additional Message Information
Message ID . . . . . : SMU16F1      Severity . . . . . : 00
Date Sent . . . . . : 12/09/94     Time Sent . . . . . : 12:39:52

Message . . . . . : Additional information received from APPN.NORA.

Cause . . . . . : Additional information was received because the activity
                  failed or the activity completed with warnings. The following information
                  was received from control point NORA with network ID APPN: MSS0218 Response
                  was sent from job 135627/MRUIZ/QCQSVSRV on APPN.NORA.
Recovery . . . . . : See the additional information received. Correct the
                  error, then try the request again.

                                                                    Bottom

Press Enter to continue.

F3=Exit  F10=Display job  F12=Cancel

```

Combining more than one activity into one change request description

After you have created a change request description in a library, you can add up to 2500 activities to that description. Those activities can have either the same action type or different action types.

One example of combined activities is running the end-of-day procedures on all systems. To add a new activity to an existing change request description, use option 1 (Add) on the Work with CRQ Description Activities display or run any of the ADDxxxCRQA commands. The xxx stands for one of these activity types:

- CMD
- OBJ
- PTF
- PRD
- RSC

The add and change request activity commands have an activity conditions (COND) parameter. If you do not change the default values, all activities are conditioned on the success of the previous activity at all nodes. Previous means the preceding activity in alphabetical order according to the activity name.

Printing the change request description

After you have created a change request description, it is a good idea to print it for documentation purposes and to verify that it is correct. To print the definition of the change request description, select option 6 (Print) on the Work with CRQ Descriptions display.

```

Work with CRQ Descriptions                               System XXXXXXXX

Type your options and press Enter.
  1=Create  2=Change  3=Copy  4=Delete  6=Print  8=Work with activities
 12=Work with submitted CRQs 13=Schedule CRQ submission 14=Submit...

Change
Request
Opt  Description  Library  Text
---  -
  1  ANLUSRPRF    GG244372  Retrieve Security Authority Users from
  2  APYPTFS      GG244372  Apply PTFs f.5763SS1, -WP1, 1234INT,
  3  DAILYWORK    GG244372  Run and Retrieve Daily reports

More...

Parameters or command
====>

F3=Exit  F4=Prompt  F6=Print list  F11=Display user and problem ID
F12=Cancel  F17=Position to  F23=More options  F24=More keys

```

Submitting the revised change request

After you have changed the request, you can use option 14 (Submit) and press F4 (Prompt), as shown in the display above, or use option 13 (Schedule CRQ submission). Option 14 starts the Submit Change Request (SBMCRQ) command and Option 13 starts the Add Job Schedule Entry (ADDJOBSCDE) command with an SBMCRQ command included or imbedded. See "Tracking submitted change requests" on page 210 for the steps to track a submitted change request.

Accessing fast path distribution commands

The basic idea of the distribution and change management functions provided by System Manager is to keep track of a series of activities that have to be performed.

The fast paths provide a single command for each activity without having to create a change request description. You can use the fast path commands by doing one of the following:

- Type the commands from any command line
- Select option 5 (Fast path distribution commands) on the System Manager main menu.
- Type:
GO FPDST

on any command line.

- Include the command or commands within a CL program, REXX procedure, or batch job stream.

Note: The fast path menu consists of more than one display; to view the second page, press the Page down key.

You can select several options from the following displays.

```

FPDST          Fast Path Distribution Commands          System: XXXXXXXX
Select one of the following:
  1. Send SMG Object
  2. Retrieve SMG Object
  3. Delete SMG Object
  4. Run SMG Object
  5. Run SMG Command

 10. Send PTF
 11. Retrieve PTF
 12. Apply Remote PTF
 13. Remove Remote PTF
 14. Delete Remote PTF

 20. Send product

                                           More ...

Selection or command
=====>
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=User support
F16=AS/400 main menu

```

On this display, SMG means System Manager. This abbreviation is used to associate the commands with the System Manager licensed program.

```

FPDST          Fast Path Distribution Commands          XXXXXXXX
Select one of the following:

 21. Retrieve product
 22. Install remote product

 40. Send license

 50. Change request administration

                                           Bottom

Selection or command
=====>
F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=User support
F16=AS/400 main menu

```

When you select an option, the associated command display is shown. For example, if you select option 2 (Retrieve SMG object), the prompt display for the Retrieve SMG Object command is shown. Specify the appropriate values, and press the Enter key to run the command.

When you use the fast path commands, change request descriptions are created and activities are added automatically. Then, the change request is submitted. After

the change request is submitted, a message is returned identifying the name of the submitted change request and a unique sequence number. The submitted change request name and sequence number can be used to track the results. You can use the Work with Submitted Change Requests (WRKSBMCRQ) command to see the results of your requests.

Note: The name of the submitted change request is always the same as the user profile that issued the fast path command. A unique sequence number is generated by the system.

The fast path commands allow you to send and retrieve PTFs on the remote system. You can also apply, remove, and delete remote PTFs.

Tip

- PTFs are applied in the order they are listed in the command.
- When you apply remote PTFs, the requisite PTFs must have already been applied on the service requester.
- When you remove remote PTFs, the dependent PTFs must have already been removed on the service requester.

For example, when you select option 13 (Remove Remote PTF), the RMVRMTPTF command display is shown. You can specify the PTF description, the destination service requester, and the node list name. You can specify other additional values by pressing F10 for additional fields (such as *Remote remove time*, *Remote IPL time*, *Time zone*, and others).

Some parameters are used for all fast paths (and also for all commands for adding activities to a CRQD), other parameters are just common to a group of commands. The prompt for the Send SMG Object (SNDSMGOBJ) command is shown in Figure 28 on page 222. The parameters applying to other fast path commands are highlighted and preceded by an ">".

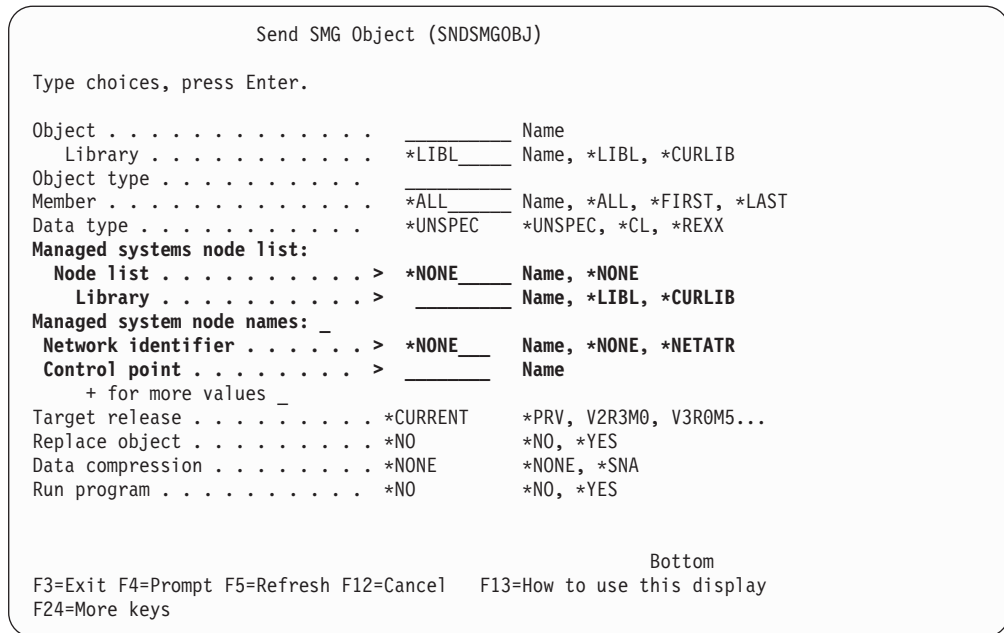


Figure 28. Send SMG Object (SNDSMGOBJ)

Note: Most parameters are not shown at the first panel when you prompt the command using F4 or the corresponding option of the menu. To see some of the additional parameters use function key F10. Others are only shown depending on the contents you entered on the first panel. To see all parameters at once, press F9, and use the page down to the parameters on the next panels.

Managed systems node list (NODL)

You can use a node list to create a group of systems with similar characteristics or just a group of all managed systems in your network. A node list is an object containing the list of systems on which the activity is to be performed. Node lists are not supported for the fast path commands for retrieve actions (RTVSMGOBJ, RTVPTE, RTVPRD). This parameter cannot be specified if the control point name (CPNAME) parameter is also specified. Only one node list can be specified for each command. The node list may also contain the name of the local system, that is the central site system.

Managed system node names (CPNAME)

Specifies the managed systems APPN control point names (CP Name) on which this request is performed. Control point (CP) names cannot be specified if a node list is specified.

The complete name of a node always consists of two parts

- The Network ID
- The CP Name

If the network ID for the central site system is the same as for the managed systems, you can use the special value *NETATR for that parameter. However, it

does not default to *NETATR, so you must always enter both parts of the name. The most convenient way to indicate the destination of your request might be the usage of node lists.

Up to 50 control point names may be specified with each fast path command (except for retrieving objects, PTFs, or products). While you might not want to enter so many names manually, this can be useful, if you plan to write programs, which retrieve those names from some database and then submit a fast path command.

Sending a license

The Send License (SNDLIC) command provides the capability to send one or more license keys from the central site system to one or more managed systems. Only the license keys matching the system serial number will be added to the managed system license repository.

A change request is automatically submitted that can be viewed to determine the status of this command. After the request is submitted, a message is returned identifying the name of the change request.

All the existing records in the central site license repository containing the license key information for the specified product will be sent to the specified managed system or systems. Only those ones that match the system serial number are added to the managed system license repository.

For more information about the license repository for the central site system and the managed system, see the *Managed System Services for AS/400 Use* book. For more information about the central site license repository, see “License information options” on page 101.

The following example sends the license keys for 1ACCOUN product with license term V3, and adds the license key matching the system serial number to the license repository of system SYS1.

```
SNDLIC PRDID(1ACCOUN)
      LICTRM(V4)
      CPNAME((*NETATR SYS1))
```

The Send Product (SNDPRD) command provides the capability to send a product previously packaged for distribution along with its license key.

The following example shows how this can be done.

Product 1ACCOUN was defined to have 2 features, each product feature has its own license key, and they are created in the central site license repository.

Table 7. Central Site License Key Repository

Product ID	License term	Feature	System Serial Number	License key
1ACCOUN	V5R2	5050	7830051	0D9432 FF85D1 AE5901
1ACCOUN	V5R2	5060	7830051	C80989 9EA0B1 903D01

Using the following command, the base part of product 1ACCOUN is installed in system SYS1. The product can be used since the license keys were sent with the

product. This command also sends the license key for the options pertaining to feature 5060 to the managed system license repository.

```
SNDPRD PRDID(1ACCOUN)
      RLS(*ONLY)
      OPTION(*BASE)
      LODTYPE(*ALL)
      TGTRLS(*CURRENT)
      CPANME((APPN SYS1))
      INSTALL(*YES)
      SNDLICKEY(*YES)
```

Supported OS/400 object types for object activities

Most OS/400 objects which can be saved and restored are supported by Managed System Services distribution services. These are the OS/400 object types supported by the object activities:

*ALRTBL	*FILE	*MSGF	*QMFORM
*BNDDIR	*FNTRSC	*MSGQ	*QMQRV
*CLD	*FORMDF	*NODL	*QRYDFN
*CLS	*FTR	*ORTABLE	*SBSD
*CMD	*GSS	*OUTQ	*SCHIDX
*CRQD	*JOBQ	*OVL	*SRVPGM
*CSI	*JOBQ	*PAGDFN	*SSND
*CSPMAP	*LIB	*PAGSEG	*TBL
*CSPTBL	*LOCALE	*PDG	*USRIDX
*DTAARA	*MENU	*PGM	*USRSPC
*FCT	*MODULE	*PNLGRP	*WSCST

Tip

To group multiple objects together or to include objects not supported by Managed System Services distribution services, use save files.

- Create a save file
- Save the objects to the save file
- Send the save file to the managed system
- Restore the objects to the target library

The object type *FILE includes all OS/400 file types, that is

- Physical files, can be sent as *FILE or *FILEDATA.
- Source physical files, can be sent as *FILE or *FILEDATA

The fast path commands as well as the ADDOBJCRQA command are part of the System Manager licensed program. The actual distribution, (for example: sending and receiving) is done by the Managed System Services licensed program. For that reason the level of support provided by Managed System Services determines which object types are supported.

- Logical files, only the description will be sent or retrieved, the underlying physical files must exist on the target system.

- Save files, can be used as *container* to transport other objects.
- Device files, such as printer files, display files, ICF files. Device files do not contain any data, only the descriptions are sent, which are similar to the save and restore commands.

Note: The *FILEDATA object type is acceptable but only when used with System Manager and Managed System Services commands. Therefore, it is not listed in the OS/400 object type list.

See the *Managed System Services for AS/400 Use* book for more information on supported object types.

Distributing installable objects

The integrated file system distribution functions are done from a central site system. To distribute an installable object from a central site iSeries server, add an object activity to a change request description object using one of the following:

- ADDOBJCRQA command
- CHGOBJCRQA command
- QNSADDCM API (use the same global name as the one specified in the PKGINSOBJ command) (see Appendix E, “System Manager APIs and exit programs” on page 291 for API information).
- ADDCRQA command
- CHGCRQA command

The following tasks are possible with installable objects:

***SND** Send the installable object from the central site system to one or more managed systems.

***RTV** Retrieve the installable object from the specified managed system to the central site system.

***DLT** Delete the catalog entry and its corresponding installable object from the managed system or systems. If the installable object has been installed, the installed objects are not deleted.

***INS** Install the installable object at the managed system(s).

***SNDINS**

Send one or more objects packaged as an installable object (at the central site system) to the managed system and install them.

***UNINS**

Delete the object(s) that were installed in the managed system.

Neither the catalog entry nor the installable object is deleted with the *UNINS action.

Note: You need to identify the object with a component name to allow the uninstall action to work correctly. For more information about component names, see “Standard names versus global names” on page 203.

Using global names with CRQ activities

A global name is the unique name by which an object is known in a network.

When OS/400 standard names are specified in the fast path commands or in change request activities, a global name is created automatically by the system.

When you want to use global names with CRQ activities, a display similar to the following is shown:

```
                                Add Object CRQ Activity (ADD0BJCRQA)

Type choices, press Enter.

Change request description . . . . . > APUPDT  Name
Library . . . . . > QGPL      Name, *LIBL, *CURLIB
Activity . . . . . > *GEN     Name, *GEN, *LAST
Action . . . . . *SND      *SND, *RTV, *DLT, *RUN ...
Object . . . . . *GLOBAL   Name, *GLOBAL, *COMPNAME
Library . . . . . _____ Name, *LIBL, *CURLIB

                                More ...

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
```

To view all associated displays, page down.

Using component names with CRQ activities

As part of the global naming scheme, the object can be identified by the component name specified on the `COMPNAME` parameter. Component names are used to distinguish objects with a unique name. The `COMPNAME` value is only valid when `ACTION(*UNINS)` and `OBJ(*COMPNAME)` are specified.

When you want to specify component names, a display similar to the following is shown:

```

Add Object CRQ Activity (ADDOBJCRQA)

Type choices, press Enter.

Component name:
Component name token 1 . . . . . _____
Component name token 2 . . . . . _____
Component name token 3 . . . . . _____
Component name token 4 . . . . . _____
Component name token 5 . . . . . _____
Component name token 6 . . . . . _____
Component name token 7 . . . . . _____
Refresh level . . . . . *ALL ___

Managed systems node list:
Node list . . . . . *NONE ___ Name, *NONE
Library . . . . . _____ Name, *LIBL, *CURLIB
Managed systems node names:
Network identifier . . . . . *NONE ___ Name, *NONE, *NETATR
Control point . . . . . _____ Name
+ for more values _

More ...

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
F13=How to use this display F24=More keys

```

When you press F10 (Additional parameters) a display similar to the following is shown:

```

Add Object CRQ Activity (ADDOBJCRQA)

Type choices, press Enter.
Additional Parameters

Activity conditions:
Activity . . . . . _____ *Name, generic*, *NONE *PRV
Relational operator . . . . . *EQ *EQ, *GT, *LT, *LE, *NE, *GE
End code . . . . . *SUCCESS 00-99, *SUCCESS, *FAIL...
Condition mode . . . . . *ALLNODES *ALLNODES, *SAMENODE
+ for more values _

Activity start time:
Start after:
Time . . . . . *CURRENT Time, *CURRENT
Date . . . . . *CURRENT Date, *CURRENT, *NEXT
Start before:
Time . . . . . *ANY Time, *ANY, *CURRENT
Date . . . . . *ANY Date, *ANY, *CURRENT, *NEXT

Bottom

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

```

For more information about catalog entries, see “Standard names versus global names” on page 203. For more information about global names, see the *Managed System Services for AS/400 Use* book.

Commands associated with change requests

In summary, Table 8 on page 228 lists the commands associated with change requests.

Table 8. Common Change Request Tasks

Task Description	Command or Option	Related Commands	Comments
Create a node list	CRTNODL	ADDNODLE	See the online help information
		DLTNODL	
		RMVNODLE	
		WRKNODL	
		WRKNODLE	
Create a change request description	CRTCRQD	ADDOBJCRQA	See "Creating a change request description" on page 198
		ADDPRDCRQA	
		ADDPTFCRQA	
		ADDCRQA	
		ADDRSCCRQA	
		CHGCRQA	
		CHGOBJCRQA	
		CHGPRDCRQA	
CHGPTFCRQA			
Work with and manage change request descriptions and their activities	WRKCRQD	CHGCRQD	See "Adding activities to change request descriptions" on page 199
		CRTCRQD	
		DLTCRQD	
Submit a change	SBMCRQ		See "Submitting change requests" on page 208
View the status of all submitted requests	DSPSBMCRQ	DSPSBMCRQA	See the online help information
		DSPSBMCRQM	
View the status of and control submitted change requests and their activities	WRKSBMCRQ	DLTSBMCRQA	See the online help information
		ENDSBMCRQA	
		HLDSBMCRQA	
		RLSSBMCRQA	
		WRKSBMCRQA	

Cleaning the change request database

There are two ways to clean up the change request database:

- Use the Change Cleanup (CHGCLNUP) command.

The value you specify for the system journals and logs is also the value used for change requests.

- Use the Delete Submitted Change Request (DLTSBMCRQ) command.

Using commands on remote systems

Using remote commands is another way to control operations. You use remote commands to send or run any command that can be used in a batch environment on the managed systems. For example, you can create message queues or submit user-written commands to be run on the managed system.

To use remote commands, you need to:

- Start Managed System Services on all systems in the network.
- Run the Start Manager Services (STRMGRSRV) command using the (*RMTCMD) value for the SERVICE parameter on the central site system.
- Run the Start System Manager (STRSYSMGR) command on the central site system.

A remote command request is a request that comes from a central site system to a managed system. Each request describes a command to be processed on the managed system and includes some related information about the request. Using this specific information, you can tell when the request was sent and the status of the request. The output, or spooled file, shows the results of the command. The spooled file is sent back to the central site system from the managed system.

Remote commands also allow you to send a command with a user ID and password that can be encoded to provide a greater level of security. When you have commands running on remote systems, this level of security is important.

The Run System Manager Command (RUNSMGCMD) command provides the capability of sending a command from the central site system to one or more remote systems. The command is shipped with the public authority of *EXCLUDE.

To use this command, you need to specify the command you want to run. The command may be any command available on the central site system that can be used in a batch environment.

You can also specify optional parameters. They are:

- Node list. This list identifies the names of the systems on which you want to run the command.
- Node name of a managed system.
- Remote start time. This value indicates the time that the remote command will start on the managed system.
- Return output spooled files. This value indicates whether or not data is returned. If you select *YES, then all spooled files created are returned to the central site system.
- User profile
- Password for the remote user profile
- Encode value. This value indicates whether or not the command, user profile, and password are encoded when the request is sent to the remote systems.

Select option 5 (Fast path distribution commands) from the System Manager menu to access the RUNSMGCMD command. When you select option 5 (Run SMG Command) from the fast path menu, the following display is shown.

```

Run SMG Command (RUNSMGCMDCMD)

Type choices, press Enter.

Command to run . . . . . _____

Managed systems node list:
Node list . . . . . *NONE      Name, *NONE
Library . . . . . _____   Name, *LIBL, *CURLIB
Managed system node names:
Network identifier . . . . . *NONE      Name, *NONE, *NETATR
Control point . . . . . _____   Name
+ for more values

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10-Additional parameters   F12=Cancel
F13=How to use this display   F24=More keys

```

Press F10 to see the additional parameters.

The following is an example of how to use the Run SMG Command command to start subsystem QSNADS on 20 systems identified in the node list SALESOFCC. In this example, the job log information is returned if the command fails.

```
RUNSMGCMDCMD  CMD(STRSBS QSNADS)  NODL(SALESOFCC)  RTNSPLF(*FAIL)
```

Displaying output

All the returned data from a single remote command is found in one spooled file. To see this spooled file, use the Display Spooled File (DSPSPLF) command or the Work with Spooled Files (WRKSPLF) command.

Diagnosing problems with change request processing

When submitting a change request, problems may occur. Some of the problems may be:

- System Manager is not started
- Managed System Services is not started
- QSNADS subsystem is not started at the central site system
- SVDS distribution queue is incorrectly configured

System Manager is not started

When you submit a change request and a problem occurs, you receive a message stating the problem and what you can do to resolve the problem. For example, message SMU1607 states that the system cannot process the request because the System Manager function is not started. You use the Start System Manager (STRSYSMGR) command to start the System Manager program.

Managed System Services is not started

Another example of a problem is when Managed System Services is not started. The SMU162C message notifies you of this problem and states that you need to use the Start Managed System Services (STRMGDSYS) command to allow the request to start processing.

QSNADS subsystem is not started at the central site system

If the QSNADS subsystem was not started at either the central site system or the managed system, any submitted change request will stay in "running" status and no error message is sent. You can use the Display Distribution Log (DSPDSTLOG) command to verify the distribution status. You can also check the system operator message queue (QSYSOPR) for any error messages or use the Work with Distribution Queues (WRKDSTQ) command to see if any entries appear.

SVDS distribution queue is not created

Before the managed system accepts any requests sent by the central site system, the system ensures that it can send any responses. If it is not possible to send responses it will reject the request stating that the system did not find the proper distribution queue or the distribution queue is not created. To resolve this situation, you can use one of the following commands:

- Use the Work with Service Providers (WRKSRVPVD) command and select option 1 (Add) to add the central site system as a service provider. This creates the *SVDS distribution queue automatically.
- Configure Distribution Service (CFGDSTSRV) command to change the configuration of the distribution network. You can add, change, delete, and display distribution queues and other related information.
- Add Distribution Queue (ADDSTQ) command to add an entry in the distribution services queue table. Distribution queues are used to store distributions before they are sent or forwarded to other systems.

Attempt to send remote command failed

When you try to send a command to a remote system and you receive a message that it failed, you need to ensure that the remote command manager is started. To start the remote command manager, use the Start Manager Services (STRMGRSRV) command and specify *RMTCMD for the value for services to start.

Appendix A. Programming and Command Reference

| This appendix no longer contains the syntax diagrams and command descriptions
| for those commands specific to the System Manager licensed program. The
| command descriptions and syntax diagrams were moved to the Information
| Center.

Appendix B. Example setup and configuration of an OS/400 network using SNADS

When Managed System Services is installed on the service requester, the necessary SNADS configuration is done automatically. This section describes how to manually configure SNADS for the service provider and the service requester when Managed System Services is not used.

The following is an example of how to set up your network using the Systems Network Architecture distribution services (SNADS).

Using Systems Network Architecture distribution services (SNADS)

System Manager uses SNADS and object distribution to distribute PTFs from an OS/400 service provider to the service requesters for which they provide service. **SNADS** is an IBM asynchronous distribution service that defines a set of rules to receive, route, and send electronic mail in a network of systems. **Object distribution** is a function that allows a user to send source and data files, save files, job streams, spooled output files, and messages to another user, either locally or in a SNADS network.

Because SNADS is being used in conjunction with APPN networks that may span enterprise boundaries, special considerations must be given to the SNADS configuration. In particular, the names used for directory entries and routing table entries must be unique within the networks being set up. For this reason, SNADS secondary system names with system name/group set to the APPN control point name and network ID are used. Routing entries and directory entries, which use both parts of the system name/group, are then needed.

On the service requester, a job named QESTP is used to automatically receive PTFs. This job runs in the QSNADS subsystem and is automatically started when the first OS/400 service provider is defined (a service provider other than IBM service support, *IBMSRV) and when the QSNADS subsystem is started.

If you have not already set up SNADS, the following procedures provide a basic setup for distribution services and object distribution that you can use on all the systems in your network. For a more complete description of setup procedures for SNADS, see the *SNA Distribution Services* book.

Creating directory entries

The directory entries are automatically created by the WRKSRVRQS and WRKSRVPVD commands.

Setting up SNADS for object distribution

1. Type the Configure Distribution Services (CFGDSTSRV) command and press the Enter key. The Configure Distribution Services display is shown.
2. Select option 1 (Distribution queues) and press the Enter key.

If you are setting up SNADS on the service provider, you need to set up a distribution queue for each service requester to which you will be distributing

PTFs. If you are setting up SNADS on a service requester, you need to set up a distribution queue for sending distribution feedback messages to your service providers.

3. Press F6 (Add distribution queue) to create one or more SNADS distribution queues.
For example, if you are setting up the service provider and will have 10 service requesters for which your system will provide service, you need to add 10 SNADS distribution queues.
4. Type the name of the queue, the remote location name (this is the control point name for the remote system), and the remote network ID (this is the network ID for the remote system). Press the Enter key.
5. Press F12 (Cancel) when a queue for each of the remote systems has been created.
6. Press F12 (Cancel) again to return to the Configure Distribution Services display.
7. Select option 2 (Routing table) to configure routing table entries and press the Enter key. You are going to add routing table entries that provide a way for your system to send information to the remote systems.
8. Press F6 (Add routing table entry) to add a routing table entry for each of the remote systems.
If you are setting up a service provider, a routing table entry is needed for each service requester for which you are providing service. If you are setting up a service requester, a routing table entry is needed for each OS/400 service provider that supports your system.
9. Type the control point name for the remote system in the *System name* field.
10. Type the network ID for the remote system in the *Group* field.
11. Type the name of the distribution queue you created for this remote system in step 3 in the fields for each service level. For simplicity, use the same queue name for each service level. Leave the default value for the maximum number of hops and press the Enter key.
12. Press F12 (Cancel) twice to return to the Configure Distribution Services display.
13. Select option 3 (Secondary system name table) to configure the secondary name table. Press the Enter key.
14. Type your local control point name, network ID (group), and a description of this system. Press the Enter key, and then press F3 (Exit). The SNADS configuration is complete for this system.

The directory entries necessary for distributing PTFs and sending distribution feedback messages are created by the user. The following information should be included in these directory entries:

- User ID/Address, which is the system control point name and network ID
- Description, which indicates whether the entry is for a service requester or for PTF distribution
- System name/Group, which is the control point name for the system
- User profile, which is QSRVBAS for the *local* directory entry

Verifying SNA distribution services (SNADS) configuration

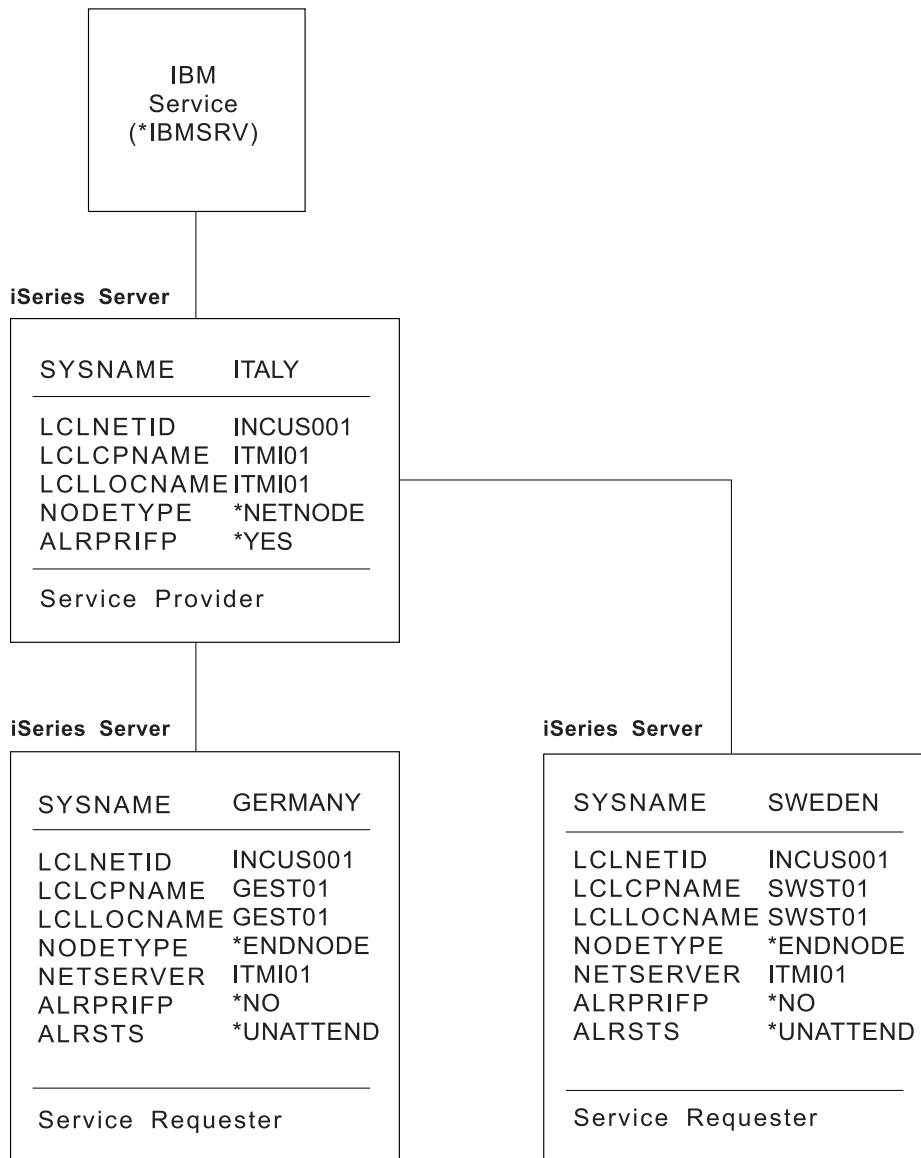
If you are presently using SNADS, the distribution queues and the routing table entries may already exist. You need to verify that the other SNADS requirements are met before you set up your service provider. This is step 3 in Figure 1 on page 7.

System Manager is dependent on the following SNADS requirements:

- A distribution queue must exist on the service provider for each service requester for which service is to be provided.
- A routing table entry must exist on the service provider to route data from the service provider to the service requesters. The routing table entries need to use the APPN control point name and network ID for the system name and group.
- A secondary system name/group that is the same as the APPN control point name/network ID for the local system must exist on all systems in the network.
- A directory entry with a user ID/address that is the same as the APPN control point name/network ID for the local system must exist on all systems. QSRVBAS should be used as the profile name.
- A directory entry for each service requester with a user ID/address that is the same as the APPN control point name/network ID must exist on the service provider.

For an example showing a SNADS setup on a service provider, see “SNADS definitions on the service provider” on page 242. For an example showing a SNADS setup on a service requester, see “SNADS definitions on the service requester” on page 249.

The following iSeries server configuration and setup example provides a more detailed look at the steps used to set up the communications and the System Manager licensed program for the network shown in Figure 29 on page 238. Use the figure for a reference as you read through the example.



RS3N014-3

Figure 29. An Example OS/400 Network

Creating directory entries on the service requester

Follow these steps to create the directory entry on the service requester:

1. Use the control point name of the local system for the CPNAME value.
2. Use the network ID of the local system for the NETID value.

Note: This step is done only once.

3.

```
ADDIRE USRID(cpname netid)
   USRD('IBM User for PTF distribution ')
   SYSNAME(*LCL)
   TEXT('Entry added by....') - example
   USER(QSRVBAS)
```


Note: This step is done for each service provider that is using *SNADS as the distribution queue.

```
ADDIRE USRID(cpname netid)
        USRD(description of service provider)
        SYSNAME(cpname netid)
        TEXT('Entry added by....') - example
```

where

- The CPNAME value is the control point name of the service provider.
- The NETID value is the network ID of the service provider.

Creating directory entries on the service provider

Follow these steps to create a directory entry on the service provider:

1. Use the control point name of the local system for the CPNAME value.
2. Use the network ID of the local system for the NETID value.

Note: This step is done only once.

```
ADDIRE USRID(cpname netid)
        USRD('IBM User for PTF distribution ')
        SYSNAME(*LCL)
        TEXT('Entry added by....') - example
        USER(QSRVBAS)
```

Note: This is done for each service requester that will be using *SNADS as the distribution queue.

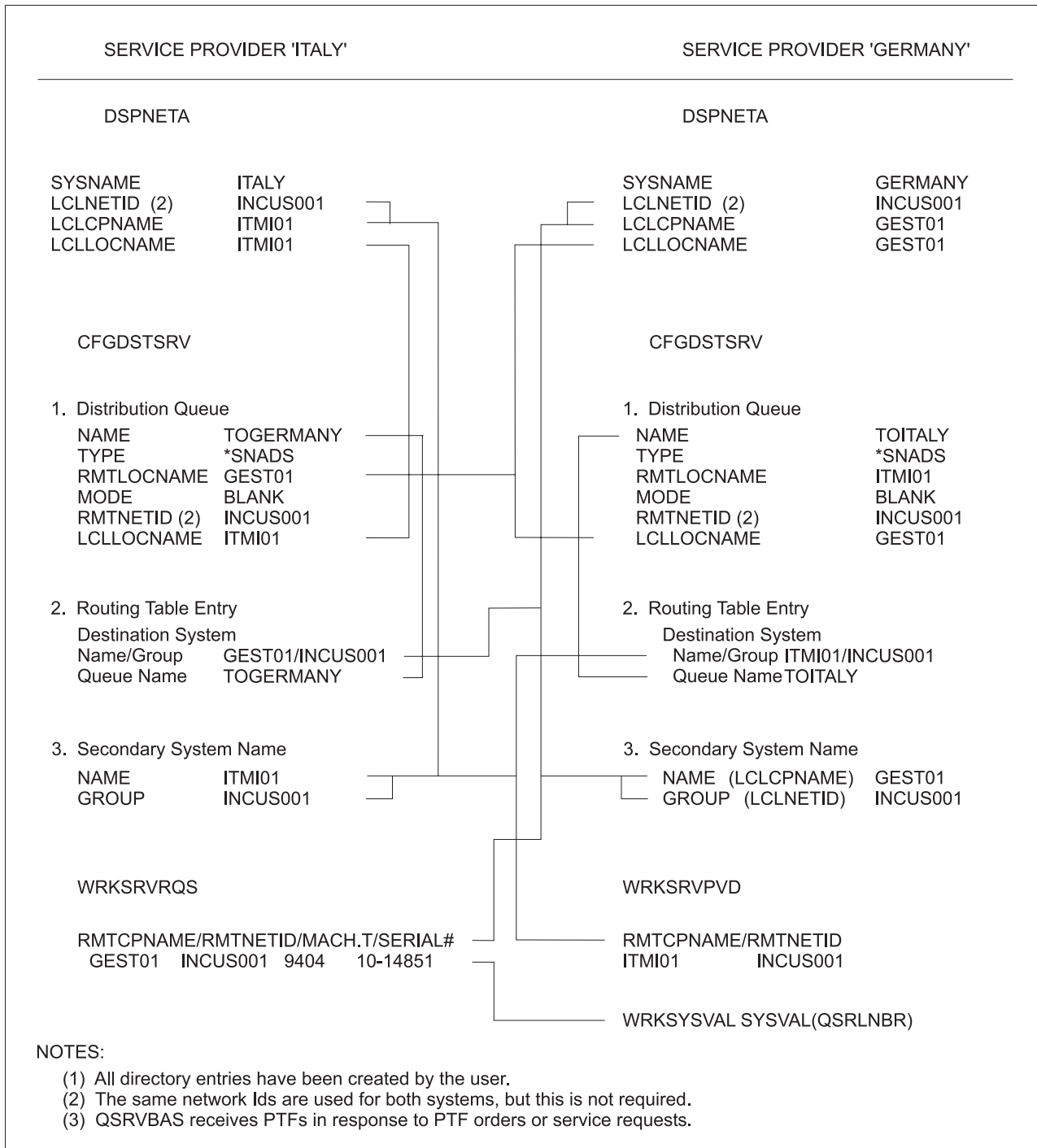
```
ADDIRE USRID(cpname netid)
        USRD(description of service requester)
        SYSNAME(cpname netid)
        TEXT('Entry added by....') - example
```

where

- The CPNAME is the control point name of the service requester.
- The NETID is the network ID of the service requester.

Matching parameter values for setting up the iSeries servers

Figure 30 on page 240 shows the commands and some of the parameter values used to set up communications and the System Manager licensed program. The lines in Figure 30 show the relationship between parameter values used to set up service provider ITALY and service requester GERMANY.



rbaep501-0

Figure 30. Matching Command Parameter Values

Setting up communications on the service provider

The example begins by showing the Change Network Attributes (CHGNETA) command and the values used for setting up system ITALY. System ITALY is the service provider for this example.

Changing the network attributes

To set the values for communications with the network and the values for alert support, the network operator for service provider ITALY does the following:

- Types the Change Network Attributes (CHGNETA) command and presses F4 (Prompt).

The following display shows the parameters and the values of the network attributes set for service provider ITALY. The numbers in parenthesis to the right of the values indicate the corresponding value on the work sheet shown in "Example work sheet" on page 254.

```
Change Network Attributes (CHGNETA)

Type choices, press Enter.

System name . . . . . SYSNAME          ITALY   (1)
Local network ID . . . . . LCLNETID     INCUS001 (2)
Local control point name . . . . . LCLCPNAME ITMI01  (3)
Default local location name . . . . . LCLLOCNAME ITMI01  (3)
Default mode . . . . . DFTMODE         BLANK
Node type . . . . . NODETYPE          *NETNODE
Data compression . . . . . DTACPR      *SAME
Intermediate data compression . . . . . DTACPRIM *SAME
Route addition resistance . . . . . RAR      128
Network node servers:          NETSERVER
  Server network ID . . . . .
  Control point name . . . . .
      + for more values
Alert status . . . . . ALRSTS          *ON
Alert logging status . . . . . ALRLOGSTS *ALL

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

The remaining network attributes are set to the default values.

Line, controller, and device descriptions on the service provider

The following commands and parameters were used to create the line, controller, and device descriptions on service provider ITALY.

Note: This is an example. Lines and controllers do not necessarily have to be configured this way.

Creating a line description

The Create Line Description (Token-Ring) (CRTLINTRN) command was used to create a line description on service provider ITALY for communications between ITALY and the two service requesters, GERMANY and SWEDEN.

The following command and parameters were used. All parameters that are not shown assume the default values.

```
CRTLINTRN LIND(DAMTRN) RSRNAME(LIN011) ADPTADR(400000000DA2)
EXCHID(05614847) SSAP((04 *MAXFRAME)(06 *MAXFRAME)(AA *MAXFRAME))
ELYTKNRLS(*NO) AUTOCRTCTL(*YES) AUTODLTCTL(*NONE)
TEXT('Line description for ITALY')
```

Creating controller descriptions

The Create Controller Description (APPC) (CRTCTLAPPC) command was used to create a controller description on service provider ITALY for communications between ITALY and service requester GERMANY.

The following command and parameters were used. All parameters that are not shown assume the default values.

```
CRTCTLAPPC CTLD(GEST01) LINKTYPE(*LAN) SWTLINLST(DAMTRN)
RMTNETID(INCUS001) RMTCPNAME(GEST01) ADPTADR(400000000DA1)
LANFRMRTY(10) LANCNNRTY(10) LANRSPTMR(10) LANCNNTMR(70)
LANACKTMR(1) LANINACTMR(100) LANACKFRQ(1) LANMAXOUT(2)
LANACCPTY(0) NODETYPE(*CALC) TMSGRPNBR(*CALC)
TEXT('APPC Controller Description for GERMANY')
```

The Create Controller Description (APPC) (CRTCTLAPPC) command was used to create a controller description on service provider ITALY for communications between ITALY and service requester SWEDEN.

The following command and parameters were used. All parameters that are not shown assume the default values.

```
CRTCTLAPPC CTLD(SWST01) LINKTYPE(*LAN) SWTLINLST(DAMTRN)
RMTNETID(INCUS001) RMTCPNAME(SWST01) ADPTADR(400000000DA3)
LANFRMRTY(10) LANCNNRTY(10) LANRSPTMR(10) LANCNNTMR(70)
LANACKTMR(1) LANINACTMR(100) LANACKFRQ(1) LANMAXOUT(2)
LANACCPTY(0) NODETYPE(*CALC) TMSGRPNBR(*CALC)
TEXT('APPC Controller Description for SWEDEN')
```

Note: The device descriptions needed are created automatically by the system.

SNADS definitions on the service provider

The following section describes the command and the options used to set up SNADS and object distribution on service provider ITALY.

Using the Configure Distribution Services (CFGDSTSRV) command

To set up SNADS on service provider ITALY, the network operator does the following:

1. On a command line, types the Configure Distribution Services (CFGDSTSRV) command and presses the Enter key.
2. Selects Option 1 (Distribution queues) and presses the Enter key.
3. Presses F6 (Add distribution queue) to add a distribution queue for service requester GERMANY.

The following values were typed:

```
Queue . . . . . TOGERMANY (4)
Queue type . . . . . *SNADS
Remote location name . . . . . GEST01 (7)
Mode . . . . . *NETATR
Remote net ID . . . . . INCUS001 (6)
Local location name . . . . . ITMI01 (3)
```

4. After the values are typed, the network operator presses the Enter key and then presses F6 (Add distribution queue) again to add a queue for service requester SWEDEN.

The following values were typed:

```
Queue . . . . . : TOSWEDEN
Queue type . . . . . : *SNADS
Remote location name . . . . : SWST01
Mode . . . . . : *NETATR
Remote net ID . . . . . : INCUS001
Local location name . . . . . : ITMI01
```

5. After the values are typed, the network operator presses the Enter key and then presses F12 (Cancel) to return to the Configure Distribution Services display.

6. From the Configure Distribution Services display, the network operator selects option 2 (Routing table entries) to create a routing table entry for each system to be designated as a service requester.
7. From the Configure Routing Table display, the network operator presses F6 (Add routing table entry).
8. On the Add Routing Table Entry display, the network operator types the values for service requester GERMANY as shown in the following list and presses the Enter key.

The values shown in parenthesis to the right of the list of values are the numbers associated with the parameter values on the work sheet shown at the end of this appendix.

```

System name/Group . . . . . GEST01  INCUS001  (7 / 6)
Description . . . . . For PTF distribution
Service level:
Fast:
  Queue name . . . . . TOGERMANY (4)
  Maximum hops . . . . . *DFT
Status:
  Queue name . . . . . TOGERMANY (4)
  Maximum hops . . . . . *DFT
Data high:
  Queue name . . . . . TOGERMANY (4)
  Maximum hops . . . . . *DFT
Data low:
  Queue name . . . . . TOGERMANY (4)
  Maximum hops . . . . . *DFT

```

9. From the Configure Routing Table display, the network operator presses F6 (Add routing table entry).
10. On the Add Routing Table Entry display, the network operator types the values for service requester SWEDEN as shown in the following list:

```

System name/Group . . . . . SWST01  INCUS001
Description . . . . . For PTF distribution
Service level:
Fast:
  Queue name . . . . . TOSWEDEN
  Maximum hops . . . . . *DFT
Status:
  Queue name . . . . . TOSWEDEN
  Maximum hops . . . . . *DFT
Data high:
  Queue name . . . . . TOSWEDEN
  Maximum hops . . . . . *DFT
Data low:
  Queue name . . . . . TOSWEDEN
  Maximum hops . . . . . *DFT

```

11. After the values are typed, the network operator presses the Enter key and then presses F12 (Cancel) to return to the Configure Distribution Services display.
12. From this display, the network operator selects option 3 (Secondary system name table) and presses the Enter key. The Configure Secondary System Name Table display is shown.
13. The network operator then types the control point name for service provider ITALY in the name field, the network ID in the group field, a brief description, and presses the Enter key. The values are shown in the following list.

```

Name: ITMI01
Group: INCUS001
Description: Entry for System Manager

```

Setting up the list of service requesters

The network operator is ready to add systems GERMANY and SWEDEN to the list of service requesters for which ITALY will provide service. The network operator does the following:

1. Types the Work with Service Requesters (WRKSRVRQS) command and presses the Enter key.
2. Types a 1 (Add) in the first *Opt* column of the display, followed by:
GEST01 (the control point name)
INCUS001 (the network ID)
Service requester GERMANY (a unique description)

and then presses the Enter key.

3. Types the contact information for service requester GERMANY on the Add Service Requester display as shown in the following displays.

```

                                Add Service Requester
                                System:  ITALY

Control point . . . . . :  GEST01      (7)
Network ID   . . . . . :  INCUS001   (6)
Description  . . . . . :  Service Requester Germany

Contact . . . . . :  Guenter Becker
Contact telephone numbers:
  Primary . . . . . :  123456
  Alternative . . . . . :  654321
Mailing address:
Street address . . . . . :  123 Hauptstrasse
City/State . . . . . :  Stuttgart
Country or Region . . . . . :  Germany
Zip code . . . . . :
Machine type . . . . . :  9406          (10)
Serial number . . . . . :  10-11011    (11)

Press Enter to continue.

F3=Exit  F5=Refresh  F12=Cancel

                                More...
```

4. Pressing the page down (roll up) key, shows the rest of the display.

```

Add Service Requester
System: ITALY

Control point . . . . . : GEST01 (7)
Network ID . . . . . : INCUS001 (6)
Description . . . . . : Service Requester Germany

Type information, press Enter.

Distribution queue type . . . . : *SNADS
Service attributes:
Log remote problems . . . . . : *SRVPVDA
Send PTFs automatically . . . . : *SRVPVDA
Message queue . . . . . : *SRVPVDA
Library . . . . . : _____
Maximum distribution . . . . . : *SRVPVDA
Cover letter language . . . . . : 2924
Forward requests automatically: *NO

Press Enter to continue.

F3=Exit F4=Prompt F5=Refresh F12=Cancel
Bottom

```

- Pressing the Enter key adds service requester GERMANY to ITALY's list of service requesters, and the same procedure is followed for service requester SWEDEN.

The service attributes shown on the display correspond to the service provider attributes set with the Change Service Provider Attributes (CHGSRVPVDA) command. The fields on the Add Service Requester display allow the network operator to set these attributes differently for individual service requesters. For this example, *SRVPVDA, the default, is used indicating the service provider values are used.

```

Add Service Requester
Control point . . . . . : SWST01
Network ID . . . . . : INCUS001
Description . . . . . : Service Requester Sweden

Contact . . . . . : Kaare
Contact telephone numbers:
Primary . . . . . : 098765
Alternative . . . . . : 567890
Mailing address:
Street address . . . . . :
City/State . . . . . : Stockholm
Country or Region . . . . . : Sweden
Zip code . . . . . :
Machine type . . . . . : 9404
Serial number . . . . . : 10-15028

Press Enter to continue.

F3=Exit F5=Refresh F12=Cancel
More...

```

```

Add Service Requester
System: ITALY

Control point . . . . . : SWST01
Network ID . . . . . : INCUS001
Description . . . . . : Service Requester Sweden

Type information, press Enter.

Distribution queue type . . . . : *SNADS
Service attributes:
Log remote problems . . . . . : *SRVPVDA
Send PTFs automatically . . . . : *SRVPVDA
Message queue . . . . . : *SRVPVDA
Library . . . . . :
Maximum distribution . . . . . : *SRVPVDA
Cover letter language . . . . . : *SRVPVDA
Forward requests automatically: *SRVPVDA

Press Enter to continue.
F3=Exit F5=Refresh F12=Cancel
Bottom

```

Creating directory entries

The following displays show the directory entries that are created by the user for service requesters GERMANY and SWEDEN on service provider ITALY when the Work with Service Requesters (WRKSRVRQS) command is used to add them as service requesters to the list of systems entitled to service from ITALY.

```

Display Directory Entry Details

User ID/Address . . . . . : GEST01 INCUS001
Description . . . . . : Service Requester GERMANY
System name/Group . . . . : GEST01 INCUS001
User profile . . . . . :
Network user ID . . . . . : GEST01 INCUS001

Name:
Last . . . . . :
First . . . . . :
Middle . . . . . :
Preferred . . . . . :
Full . . . . . :

Department . . . . . :
Job title . . . . . :
Company . . . . . :

Press Enter to continue.
F3=Exit F12=Cancel
More...

```



```

Display Directory Entry Details
User ID/Address . . . . : SWST01  INCUS001
Description . . . . . : Service Requester SWEDEN
System name/Group . . . : SWST01  INCUS001
User profile . . . . . :
Network user ID . . . . : SWST01  INCUS001

Name:
  Last . . . . . :
  First . . . . . :
  Middle . . . . . :
  Preferred . . . . . :
  Full . . . . . :

Department . . . . . :
Job title . . . . . :
Company . . . . . :

Press Enter to continue.

F3=Exit  F12=Cancel
More...

```

These entries are created by the user. They are shown here because they are necessary to the SNADS and object distribution functions and should not be deleted.

The local directory entry is created by the user. This entry is shown in the following display. This directory entry is used for PTF distribution and to allow distribution feedback messages to be returned to the service provider.

```

Display Directory Entry Details
User ID/Address . . . . : ITMI01  INCUS001
Description . . . . . : IBM User for PTF distribution
System name/Group . . . : ITMI01
User profile . . . . . : QSRVBAS
Network user ID . . . . : ITMI01

Name:
  Last . . . . . :
  First . . . . . :
  Middle . . . . . :
  Preferred . . . . . :
  Full . . . . . :

Department . . . . . :
Job title . . . . . :
Company . . . . . :

Press Enter to continue.

F3=Exit  F12=Cancel
More...

```

Setting up communications on the service requester

The system operator for service requester GERMANY must use some of the same commands and values used for setting up service provider ITALY. GERMANY is to be set up as a service requester.

Changing the network attributes for a service requester

The Change Network Attributes (CHGNETA) command is used to set the values for communications with the network and the values for alert support for service requester GERMANY. Because GERMANY is an end node and ITALY is the network node server, alerts are sent to service provider ITALY. The system operator types the Change Network Attributes (CHGNETA) command and presses F4 (Prompt).

The following display shows the parameters and the values of the network attributes set for service requester GERMANY. The numbers in parenthesis to the right of the values indicate the corresponding value on the work sheet shown in "Example work sheet" on page 254.

```
Change Network Attributes (CHGNETA)

Type choices, press Enter.

System name . . . . . SYSNAME          GERMANY (5)
Local network ID . . . . . LCLNETID     INCUS001 (6)
Local control point name . . . . . LCLCPNAME  GEST01 (7)
Default local location name . . . . . LCLLOCNAME  GEST01 (7)
Default mode . . . . . DFTMODE          BLANK
Node type . . . . . NODETYPE            *ENDNODE
Data compression . . . . . DTACPR        *SAME
Intermediate data compression . . . . . DTACPRIM  *SAME
Route addition resistance . . . . . RAR      128
Network node servers:      NETSERVER     INCUS001 ITMI01 (2/3)
  Server network ID . . . . .
  Control point name . . . . .
                                + for more values
Alert status . . . . . ALRSTS            *UNATTEND
Alert logging status . . . . . ALRLOGSTS   *NONE

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

The remaining network attributes for service requester GERMANY are set to the default values.

Line, controller, and device descriptions on service requesters

Creating a line description

The Create Line Description (Token-Ring) (CRTLINTRN) command was used to create a line description on service requester GERMANY for communications between GERMANY and service provider ITALY.

The following command and parameters were used. All parameters that are not shown assume the default values.

```
CRTLINTRN LIND(DAMTRN) RSRNAME(LIN011) ADPTADR(400000000DA1)
EXCHID(05614851) SSAP((04 *MAXFRAME)(06 *MAXFRAME)(AA *MAXFRAME))
ELYTKNRLS(*NO) AUTOCRTCTL(*YES) AUTODLTCTL(*NONE) CMNRCYLMT(1 1)
TEXT('Line description for system GERMANY')
```

Creating a controller description

The Create Controller Description (APPC) (CRTCTLAPPC) command was used to create a controller description on service requester GERMANY for communications between GERMANY and service provider ITALY.

The following command listing shows the parameters that were used. All parameters that are not shown assume the default values.

```

CRTCTLAPPC CTLD(ITMI01) LINKTYPE(*TRLAN) SWTLINLST(DAMTRN)
RMTNETID(*NETATR) RMTCPNAME(ITMI01) SWTDSC(*NO)
ADPTADR(400000000DA2) TRNFRMRTY(10) TRNCNNRTY(10)
TRNRSPTMR(10) TRNCNNTMR(70) TRNACKTMR(1) TRNINACTMR(100)
TRNACKFRQ(1) TRNMAXOUT(2) TRNACCPY(0) NODETYPE(*NETNODE)
CMNRCYLMT(2 5) TEXT('APPC Controller for ITALY')

```

Note: The device descriptions needed are created automatically by the system.

SNADS definitions on the service requester

The following section describes the command and the options used to set up SNADS and object distribution for the example network shown in Figure 29 on page 238. SNADS allows the service requester to receive PTFs and return acknowledgments to the service provider.

Using the Configure Distribution Services (CFGDSTSRV) command

The values shown in parenthesis to the right of the list of values are the numbers associated with the parameter values on the work sheet shown at the end of this appendix.

The system operator at service requester GERMANY does the following:

1. Types the Configure Distribution Services (CFGDSTSRV) command on a command line and presses the Enter key.
2. Selects option 1 (Distribution queues) and presses the Enter key.
3. Presses F6 (Add distribution queue) to add a distribution queue for system ITALY.

The following values were typed:

```

Queue . . . . . TOITALY (8)
Queue type . . . . . *SNADS
Remote location name . . . . . ITMI01 (3)
Mode . . . . . *NETATR
Remote net ID . . . . . INCUS001 (2)
Local location name . . . . . GEST01 (7)

```

4. After the values are typed, the system operator presses the Enter key and then presses F12 (Cancel) to return to the Configure Distribution Services display.
5. From the Configure Distribution Services display, the system operator selects option 2 (Routing table entries) to create a routing table entry for service provider ITALY.
6. From the Configure Routing Table display, the system operator presses F6 (Add routing table entry).
7. On the Add Routing Table Entry display, the system operator types the values for service provider ITALY as shown in the following list and presses the Enter key.

```

System name/Group . . . . . ITMI01 INCUS001 (3/2)
Description . . . . . For PTF distribution
Service level:
Fast:
Queue name . . . . . TOITALY (8)
Maximum hops . . . . . *DFT
Status:
Queue name . . . . . TOITALY (8)
Maximum hops . . . . . *DFT
Data high:
Queue name . . . . . TOITALY (8)
Maximum hops . . . . . *DFT

```

```

Data low:
Queue name . . . . . TOITALY          (8)
Maximum hops . . . . . *DFT

```

8. After these values are typed, the system operator presses the Enter key, and then presses F12 (Cancel) to return to the Configure Distribution Services display.
9. From this display, option 3 (Secondary system name table) is selected and the Configure Secondary System Name Table display is shown.

Note: This part of the SNADS definitions is required for System Manager.

10. The operator types the control point name for service requester GERMANY in the name field, the network ID in the group field, a brief description, and presses the Enter key. The values are shown in the following list:

```

Name: GEST01
Group: INCUS001
Description: Entry for System Manager

```

Setting up the service requester for OS/400 service provider support

Once communications, SNADS, and object distribution are set up on service requester GERMANY, the system operator can begin the procedures to set up the iSeries server as a service requester. The system operator uses the commands provided with the Operating System/400 licensed program to create a list of service providers and to set up a user profile that allows the network operator at service provider ITALY to run remote problem analysis.

Setting up the list of service providers

The system operator begins setting up service requester GERMANY by creating a list of service providers using the Work with Service Providers (WRKSRVPVD) command. In the example, there is only one OS/400 service provider, which is also the route to IBM service support (*IBMSRV). The numbers in parenthesis to the right of the parameter values indicate the corresponding value on the work sheet shown at the end of the example.

The system operator does the following:

1. Types the Work with Service Providers (WRKSRVPVD) command and presses the Enter key.
2. Types a 1 (Add) in the first *Opt* column on the display, followed by:
 - ITMI01 (the control point name) (3)
 - INCUS001 (the network ID) (2)
 - Service Provider Italy
and then presses the Enter key.
3. Types the contact information for ITALY on the Add Service Provider display as shown below.

```

                                Add Service Provider
                                System:  GERMANY
Control point . . . . . : ITMI01
Network ID . . . . . : INCUS001
Description . . . . . : Service Provider Italy

Type information, press Enter.

Contact . . . . . Giorgio
Contact telephone numbers:
  Primary . . . . . 18005551234
  Alternative . . . . . 18005554321
Mailing address:
  Street address . . . . . 123 Via Pisano
                               _____
                               _____
  City/State . . . . . Florence
  Country or Region . . . . . Italy
  Zip code . . . . . 55555

                                Bottom
F3=Exit  F5=Refresh  F12=Cancel

```

4. Presses the Enter key and returns to the Work with Service Providers display.
5. To change the route to IBM service support (*IBMSRV), the system operator presses F22 (Change IBM service route) and the Change IBM Service Route display is shown.
 - Types ITMI01 in the control point name field.
 - Types INCUS001 in the network ID field.

When the Enter key is pressed, the Work with Service Providers display is shown with the *IBMSRV entry removed from the list. The system operator presses F3 (Exit) to leave the Work with Service Providers display.

Creating directory entries

Two directory entries are automatically created on service requester GERMANY when the operator uses the Work with Service Providers (WRKSRVPVD) command to set up the list of service providers. One directory entry is created to allow GERMANY to send service requests, PTF orders, and other communications to service provider ITALY. The second directory entry created is a local entry that is used by service requester GERMANY to send distribution feedback messages to service provider ITALY. The values used for the first directory entry are shown in the following display.

```

Display Directory Entry Details
User ID/Address . . . . : ITMI01  INCUS001  (3/2)
Description . . . . . : Service Provider ITALY
System name/Group . . . : ITMI01  INCUS001
User profile . . . . . :
Network user ID . . . . : ITMI01  INCUS001

Name:
  Last . . . . . :
  First . . . . . :
  Middle . . . . . :
  Preferred . . . . . :
  Full . . . . . :

Department . . . . . :
Job title . . . . . :
Company . . . . . :

Press Enter to continue.

F3=Exit  F12=Cancel  F18=Display location details
More...

```

The following display shows the other directory entry created for service requester GERMANY on system GERMANY when using the Work with Service Providers (WRKSRVPVD) command to add ITALY to the list of service providers.

```

Display Directory Entry Details
User ID/Address . . . . : GEST01  INCUS001  (7/6)
Description . . . . . : IBM User for PTF distribution
System name/Group . . . : GEST01
User profile . . . . . : QSRVBAS
Network user ID . . . . : GEST01

Name:
  Last . . . . . :
  First . . . . . :
  Middle . . . . . :
  Preferred . . . . . :
  Full . . . . . :

Department . . . . . :
Job title . . . . . :
Company . . . . . :

Press Enter to continue.

F3=Exit  F12=Cancel  F18=Display location details
More...

```

Both directory entries are necessary to the SNADS and object distribution functions and should not be deleted.

Creating a user profile for remote problem analysis

A user profile needs to be created on service requester GERMANY so that the network operator at service provider ITALY can sign on and run remote problem analysis. The following display shows the simplest user profile that can be used for running remote problem analysis.

```

                                Create User Profile (CRTUSRPRF)

Type choices, press Enter.

User profile . . . . .      RMTPA      Name      (9)
User password . . . . .    RMTPA      Name, *USRPRF, *NONE
Set password to expired . . *NO      *NO, *YES
Status . . . . .          *ENABLED *ENABLED, *DISABLED
User class . . . . .      *SYSOPR *USER, *SYSOPR, *PGMR...
Assistance level . . . . . *SYSVAL *SYSVAL, *BASIC, *INTERMED...
Current library . . . . . *CRTDFT Name, *CRTDFT
Initial program to call . . *NONE    Name, *NONE
Library . . . . .         Name, *LIBL, *CURLIB
Initial menu . . . . .    MAIN     Name, *SIGNOFF
Library . . . . .         *LIBL   Name, *LIBL, *CURLIB
Limit capabilities . . . . . *NO      *NO, *PARTIAL, *YES
Text 'description' . . . . . User profile for remote problem analysis

                                Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Granting object authority on the service requester

The system operator on service requester GERMANY types the following commands to grant authority to user RMTPA (the network operator on service provider ITALY) to use the necessary problem analysis commands:

- GRTOBJAUT OBJ(WRKPRB) OBJTYPE(*CMD) USER(RMTPA)
- GRTOBJAUT OBJ(ANZPRB) OBJTYPE(*CMD) USER(RMTPA)
- GRTOBJAUT OBJ(PRTERLOG) OBJTYPE(*CMD) USER(RMTPA)

Changing system values on the service requester

The final step in setting up service requester GERMANY is to change the system values for remote sign-on (QRMTSIGN) and automatic virtual device configuration (QAUTOVRT). Using the Change System Value (CHGSYSVAL) command, the system security officer changes these by typing the following on a command line and pressing the Enter key:

- CHGSYSVAL SYSVAL(QRMTSIGN) VALUE(*VERIFY)
- CHGSYSVAL SYSVAL(QAUTOVRT) VALUE(20)

The network operator at service provider ITALY is now able to sign on, run remote problem analysis, and distribute PTFs to service requester GERMANY.

Setting up another service requester

The setup procedures for service requester SWEDEN are the same as those used to set up service requester GERMANY. The system operator for SWEDEN uses the same commands and parameter values with the following exceptions:

```

System name: SWEDEN
Control point name: SWST01
Local location name: SWST01

```

When the setup for service requester SWEDEN is complete, the network operator at service provider ITALY can begin using System Manager functions.

Example work sheet

The following table shows the work sheet used to set up service provider ITALY and service requester GERMANY.

Value	Data	Parameter	Parameter Description
Service Provider			
ITALY (1)	LCLSYSNAME SYSNAME	Current System Name System Name	
INCUS001 (2)	LCLNETID SECSYSGROUP RMTNETID ADDRESS GROUP DSTSISGROUP	Local Network ID Secondary System Group Remote Network ID User Address Group Name Destination System Group	
ITMI01 (3)	LCLCPNAME SECSYSNAME LCLLOCNAME USERID SYSNAME RMTLOCNAME RMTCPNAME DSTSISNAME	Local Control Point Name Secondary System Name Local Location Name User ID System Name Remote Location Name Remote Control Point Destination System Name	
TOGERMANY (4)	DSTQNAME	Distribution Queue Name	
	Service Requester		
GERMANY (5)	LCLSYSNAME SYSNAME	Current System Name System Name	
INCUS001 (6)	LCLNETID SECSYSGROUP RMTNETID ADDRESS GROUP DSTSISGROUP	Local Network ID Secondary System Group Remote Network ID User Address Group Name Destination System Group	
GEST01 (7)	LCLCPNAME SECSYSNAME LCLLOCNAME USERID SYSNAME RMTLOCNAME RMTCPNAME DSTSISNAME	Local Control Point Name Secondary System Name Local Location Name User ID System Name Remote Location Name Remote Control Point Destination System Name	
8	TOITALY	DSTQNAME	Distribution Queue Name

Value	Data	Parameter	Parameter Description
RMTPA/RMTPA (9)	USRPRF	Problem Analysis User Profile	
10	9406		Machine Type
11	10-11011		Serial Number

Appendix C. Examples of problem situations

The problem situations described in this chapter are provided to help you build a conceptual view of the way the functions provided by System Manager combine to assist you with problem management in a network environment.

The functions described in this manual apply not only to IBM licensed programs, but also to applications packaged using the System Manager program.

Depending on your network environment and the severity of a problem, you should be able to handle problems from many iSeries servers with a single service provider. For a very large number of systems, a single operator at a service provider will not be able to effectively handle the number of problem notifications that could possibly come in at one time.

If you are an IBM business partner with a large number of customers who have networks, or an IBM customer with a large number of iSeries servers, it is a good idea to encourage your customers to use the System Manager licensed program to manage their own network problems from a service provider that reports to your system as the next-level service provider. The setup for your system is the same as for any other service provider, and your service requesters are the service providers for your customer's networks. Depending on your service contract with IBM, you can have your service requesters report all their problems to your system or you can have them report only software problems to your system, and report hardware problems to IBM service support (*IBMSRV). No matter how you set up your systems, System Manager can help you design a maintenance and service strategy that meets your needs. The following examples will help you understand how the problem management and change management functions work in different environments.

General assumptions about attributes

For the sake of simplicity, some assumptions have been made, such as the values of certain network attributes and service provider attributes. Attribute values that are specific to an example are shown with the example. The general assumptions that apply to all the examples are shown in the following list:

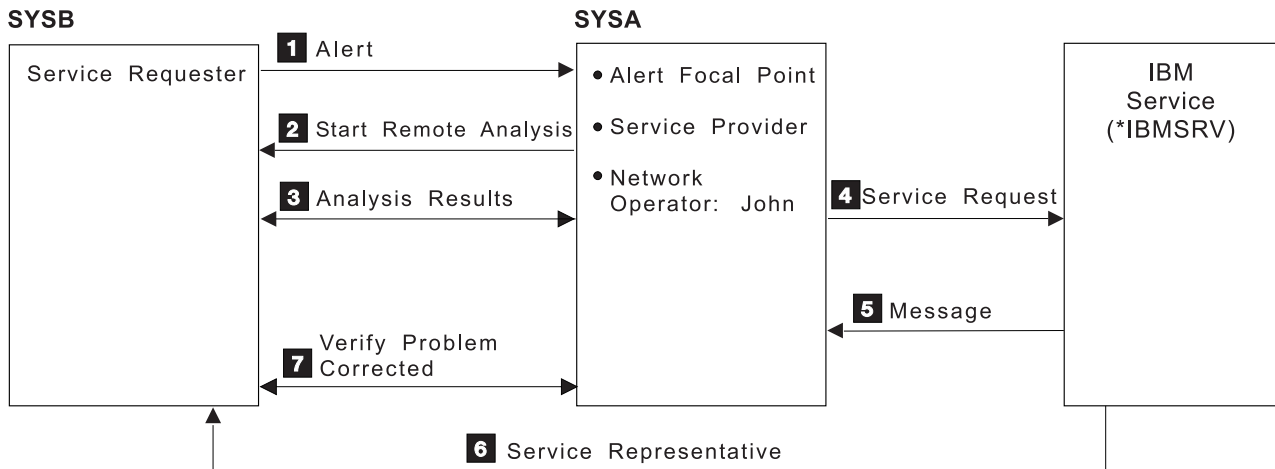
- The System Manager program is installed on the OS/400 service provider.
- The service requester is included in the list of systems for which the service provider provides service.
- The service provider is in the list of service providers for this service requester.
- The service provider message queue is named NETOPR and is in library QGPL.

Alert reported a system-detected problem

The following specific assumptions are made for this problem situation:

- The log remote problems (LOGRMTPRB) parameter of the Change Service Provider Attributes (CHGSRVPVDA) command is set to *ALERT or *ALL.
- The service provider is an alert focal point, and the alert logging status (ALRLOGSTS) parameter of the Change Network Attributes (CHGNETA) command is set to *RCV.

- The service requester alert status (ALRSTS) parameter of the Change Network Attributes (CHGNETA) command is set to *UNATTEND, and the service requester is in the sphere of control of the service provider.
- The service requester is not maintained by a full-time operator.
- The name of the network operator for the service provider is John Smith.



RS3N008-5

Figure 31. Communications Flow for Alert Reported Problem

When service requester SYSB detects a problem, both an alert and a problem record are created. The alert is then sent over a communications session to the service provider. (See 1 in Figure 31.) When the alert is received at the service provider, the alert is logged, a problem record is created in OPENED status, and a message is sent to the NETOPR message queue.

John Smith, the operator at the service provider, checks NETOPR periodically through the day to look for incoming problems and PTF orders. The message from the alert is there and John notes that the message is preceded by an asterisk (*). He presses F14 (Run problem analysis), and the Work with Problem display is shown. He then types an 8 to work with the problem. From the Work with Problem menu, he selects the option to analyze the problem and remote problem analysis is started on the service requester. (See 2 in Figure 31.)

John knows that no one is available to assist him at the service requester, and he knows that he cannot complete problem analysis if an operator is needed. When the Operator Intervention display is shown, he selects the option indicating no operator is present.

No operator is needed, so John is able to complete the analysis. The problem status is now READY on the service provider and SENT on the service requester. The results of the problem analysis, which indicate that a storage card has failed, are shown in a possible causes list. (See 3 in Figure 31.)

Because the problem involves hardware, John knows he needs to report the problem to IBM service support (*IBMSRV). (See 4 in Figure 31.) From the possible causes list, he presses F6 (Prepare service request) to report the problem. He checks the contact information displayed to verify that it contains the information the service representative needs to contact him before going to the site where the service requester is located. John then selects a severity for the problem, and selects

IBM service support (*IBMSRV) as the destination for the request. He sends the request immediately and a message is returned indicating a service representative will call. (See 5 Figure 31 on page 258.)

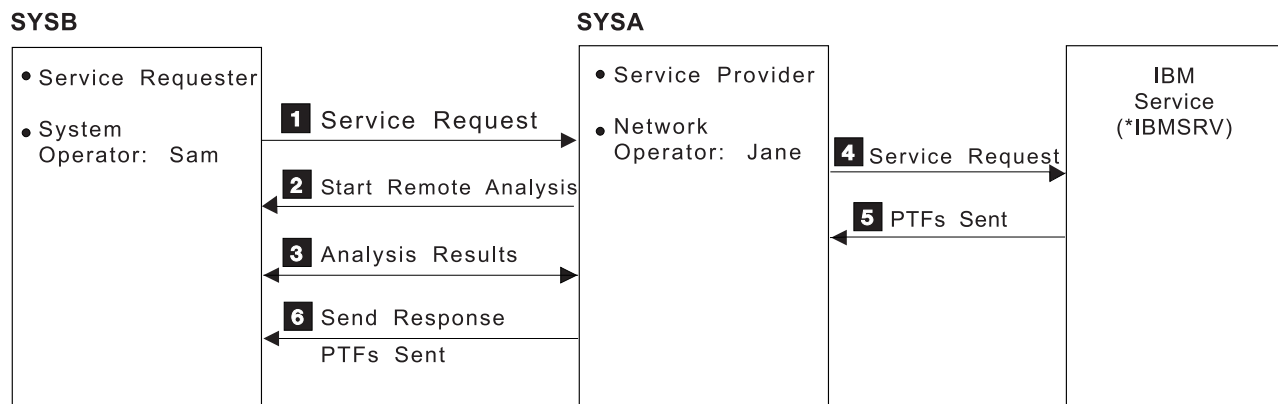
Soon, John receives a call from IBM service support letting him know that the service representative has the replacement storage card indicated in the possible causes list sent with the service request. John arranges to meet the service representative at the location of the service requester. (See 6 Figure 31 on page 258.)

When the new storage card is installed, John uses the Work with Problem (WRKPRB) command on the service requester to show the problem log and to work with the problem. He selects the option to verify the problem has been corrected and then selects the option to run verification procedures. The verification procedures run successfully and John is on his way back to the central site service provider where he indicates the problem has been verified as corrected. (See 7 Figure 31 on page 258.)

Service request reported a system-detected problem

The following assumptions are made for this problem situation:

- The log remote problems (LOGRMTPRB) parameter of the Change Service Provider Attributes (CHGSRVPVDA) command is set to *SRVRQS or *ALL.
- The send PTFs automatically (SNDPTFAUTO) parameter of the Change Service Provider Attributes (CHGSRVPRVA) command is set to *YES.
- The service requester is maintained by an operator, Sam Jones, who is available during one shift.
- The name of the network operator for the service provider is Jane Brown.



RS3N006-4

Figure 32. Service Request Reported System-Detected Problem

Sam Jones, an operator for an OS/400 service requester, checks the QSYSOPR message queue once or twice a day. This morning, Sam finds a message indicating that one of his communications lines is down. The message is preceded by an asterisk (*) so he checks the problem log to find the corresponding problem record.

Sam is not experienced with communications, so he decides to report the problem immediately. He types the option to work with the problem and then selects the option to report the problem. Sam checks the contact information, selects a severity of 2, and selects SYSA as the destination of his service request. The service request is sent to Jane's server. (See 1 Figure 32.)

When the service request arrives at the service provider, a problem record is created in OPENED status, and a message is sent to the service provider message queue, NETOPR. Jane, who checks NETOPR regularly, sees the message preceded by an asterisk (*), moves the cursor to it and presses the function key to run problem analysis. The Work with Problem display is shown and she types an 8 next to the problem sent by Sam. The Work with Problem menu is then shown for Jane to select the option to analyze the problem. (See 2 Figure 32 on page 259.)

Because the problem is with a communications line, Jane is asked to indicate whether or not an operator is available at the remote system to assist her with the analysis. She presses the function key provided to look at the contact information for the service requester and finds that Sam should be there at this time of day. She calls Sam on the telephone so he is available to assist with the analysis. When Jane has finished signing on to the service requester, she has Sam type the Work with Active Jobs (WRKACTJOB) command to look for the device name being used for the display station pass-through session she is using to run the analysis. Next, she has Sam type the Start Copy Screen (STRCPYSCN) command specifying the device name for the display station pass-through session as the source device.

Jane proceeds with the analysis while Sam observes and assists her with things like checking cables, controller cards, and the modem. The analysis results indicate the problem is with Licensed Internal Code related to communications, and the possible causes list is shown on both displays. The analysis results are used to update the problem records on both systems. (See 3 Figure 32 on page 259.)

When analysis is complete, the PTF database is searched for PTFs that correct communications problems. No PTFs are found that resolve this problem, so Jane selects the option to report the problem. When the contact information is displayed, she presses the function key provided to use the local contact information rather than the contact information for the service requester. This ensures that the answering PTFs are sent to the service provider. Jane then specifies a severity of medium and indicates the destination of her service request is IBM service support (*IBMSRV). She sends the request immediately. (See 4 Figure 32 on page 259.)

After a short wait, a Licensed Internal Code fix is returned along with some required PTFs. (See 5 Figure 32 on page 259.) Jane selects the option to send a response to Sam's system and the new PTFs are sent to Sam's service requester immediately. (See 6 Figure 32 on page 259.) Sam loads and applies the PTFs as soon as he receives them and then checks to see that his communications problem is resolved.

Jane is concerned that she was not able to test the PTFs before distributing them to Sam's system, so she places the new PTFs on hold (HLDPTF command) and leaves the problem record in ANSWERED status for a week to allow Sam to verify whether or not problems occur that are related to the PTFs. After a week, Sam contacts Jane to let her know that no more problems have occurred and Jane releases the PTFs so they can be distributed.

User-detected performance problem

The following assumptions are made for this problem situation:

- The log remote problems (LOGRMTPRB) parameter of the Change Service Provider Attributes (CHGSRVPVDA) command is set to *SRVRQS or *ALL.
- The send PTFs automatically (SNDPTFAUTO) parameter of the Change Service Provider Attributes (CHGSRVPVDA) command is set to *NO.

- The service requester is maintained by a help desk clerk named Sally, who has limited experience with system operations.
- The name of the network operator for the service provider is John Smith.

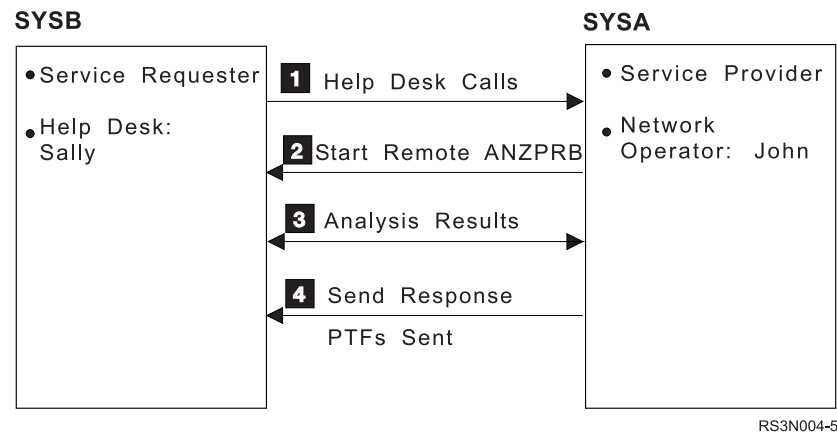


Figure 33. A User-Detected Performance Problem

John Smith is working on problem log maintenance this morning. He is working on deleting the problems that are over 30 days old, when he receives a telephone call from Sally, a help desk clerk for one of his service requester. Sally tells John that some of the users on her system are complaining that the system is very slow to respond. Tasks that normally take a few seconds, are taking up to 30 seconds to process. She asks if John has time to check on the problem. (See 1 Figure 33.)

John decides to use the Analyze Problem (ANZPRB) command specifying remote analysis, the service requester control point name, and network ID. (See 2 Figure 33.) As he is analyzing the problem, John finds the system responding very slowly. After a while, he completes analysis by creating a problem record and uses the Work with Problem (WRKPRB) command to look at the details for the problem he just analyzed. (See 3 Figure 33.)

When problem analysis is complete, the system searches for PTFs that correct performance problems and a PTF to adjust system performance is found along with some requisite PTFs. All the PTFs needed to correct the problem are on the service provider; however, two of the PTFs needed are on hold for testing. John needs to find out whether the testing is complete for these two PTFs, so he uses the Display PTF (DSPPTF) command to check whether the PTFs have been loaded and applied on the system for testing. He finds that the two PTFs have been temporarily applied and have been on his system for just over a week with no problems reported. John then uses the Release PTF (RLSPTF) command to release them.

John is now ready to distribute the PTFs to Sally's service requester. He uses the Work with Problem (WRKPRB) command to display the problem log and types the option to work with the problem. He then selects the option to send a response to the service requester and the PTFs are sent. (See 4 Figure 33.) John calls Sally at the help desk to let her know that he or one of his co-workers will come to the service requester site later in the day to load and apply the PTFs.

Once the PTFs are installed, John checks with Sally during the following week to be sure the service requester is running properly and the slow response does not occur again.

PTF order and distribution

The following assumptions are made for this problem situation:

- The log remote problems (LOGRMTPRB) parameter of the Change Service Provider Attributes (CHGSRVPVDA) command is set to *SRVRQS or *ALL.
- The send PTFs automatically (SNDPTFAUTO) parameter of the Change Service Provider Attributes (CHGSRVPVDA) command is set to *YES.
- The service requester is maintained by an operator, Sam Jones, who is available during one shift.
- The name of the network operator for the service provider is Jane Brown.

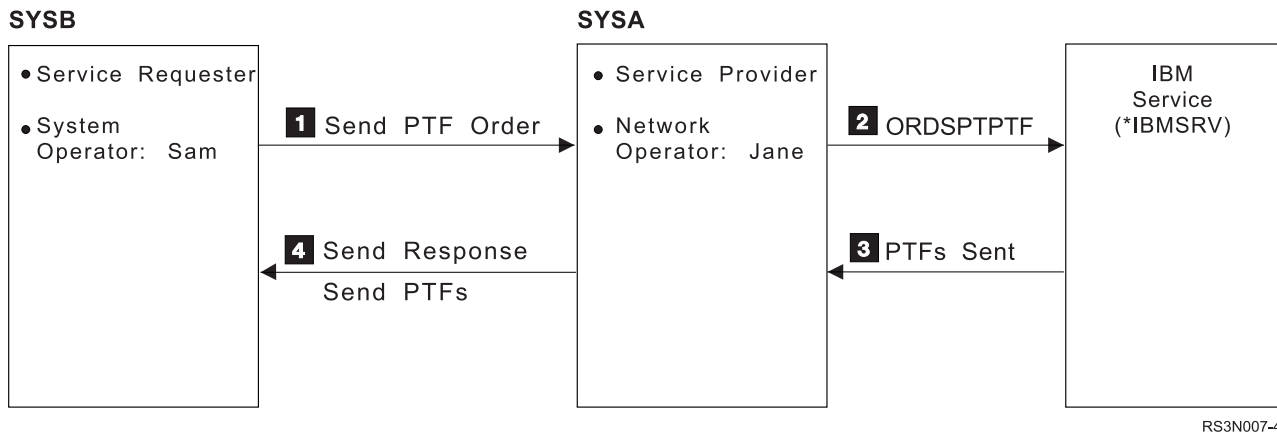


Figure 34. A Send PTF Order and Distribution

Jane Brown tries to keep her network at the most current maintenance level; however, she also allows the operators at her service requester to determine the needs for their particular systems. Every month she orders the latest PTF summary list and distributes it to all her service requesters. Jane also uses the Order Supported Product PTFs (ORDSPTPTF) command regularly to be sure she has the latest PTFs available for the licensed programs she supports.

Sam, an operator at one of the service requesters in Jane's network, has been looking over the PTF summary list and decides he should order the PTFs available for the new licensed program his company recently purchased. Using the Send PTF Order (SNDPTFORD) command, he specifies the licensed program number and the applicable PTF numbers from the PTF summary list. He then sends the order to his service provider. (See 1 Figure 34.)

At the service provider, Jane has been away from her desk for a while. When she returns, she decides to check NETOPR to see if any new messages have come in. She sees Sam's PTF order, which has recently arrived, and that her system does not provide service support for the new licensed program purchased by Sam's company.

Jane types the Work with Supported Products (WRKSPTPRD) command and presses the function key provided to support additional products. This shows her a list of the products and product options for which she does not currently provide service support. She pages down through the list until she finds the new licensed program, and types the option to add support. She also decides to add support for all the associated product options as well. Jane returns to the Work with Supported Products display and recalls that Sam's company has a branch office in Portugal, so she decides to add language feature support for the new licensed program. She

types the option to work with supported language features in the option column of the licensed program and each of the product options and then presses the function key provided to add language feature support. Jane selects Portuguese and leaves the Work with Supported Products function.

In order to provide effective support for the new licensed program, Jane decides to use the Order Supported Product PTFs (ORDSPTPTF) command to order all the PTFs currently available. She specifies *LINKONLY for the delivery method and *ALL for the licensed program number so that any necessary requisites, such as Licensed Internal Code PTFs or Operating System/400 PTFs, are also returned with her order. She then presses the Enter key to send the request. (See 2 Figure 34 on page 262.)

The PTF order is large; however, it does not exceed the transmission limit. While checking the NETOPR message queue, Jane sees a notification about a defective PTF that she currently has released and made available for distribution. She places this PTF on hold immediately, using the Hold PTF (HLDPTF) command, so that the PTF can no longer be distributed. This gives her time to find out more about the defective PTF.

In about an hour, she has all the necessary PTFs available for the new licensed program. (See 3 Figure 34 on page 262.) Jane then uses the Work with Problem (WRKPRB) command to find and work with the problem record for the PTF order from Sam. She selects the option to send a response and the PTFs Sam ordered are sent immediately. (See 4 Figure 34 on page 262.)

Because the PTFs Jane ordered are new, she notes this in the problem record notes and decides not to close the PTF order until Sam lets her know whether the PTFs cause any problems on his system.

Detailed examples of problem situations

The examples presented in this section provide detailed steps and information to guide you through the types of situations that occur when you use System Manager to perform problem management and change management for your own network.

Illustrations are presented first to show the direction of communications flow between the service requester and service provider, followed by an explanation of the problem situation. The general assumptions for the previous examples are also used for the detailed examples. The numbers in the illustrations correspond to the numbers used in the problem situation description.

Hardware service request report

In this example, an OS/400 service requester is having a problem with a diskette unit. The following specific assumptions are made for this problem situation:

- The alert status (ALRSTS) network attribute is set to *UNATTEND.
- The network operator monitors the service provider message queue and regularly checks the Work with Alerts display.
- The system operator at the service requester is present, but has little experience with problem management.

Figure 35 on page 264 describes the steps used to work with this problem.

Service Requester			Service Provider		Next Service Provider System
Action	Status	↔	Action	Status	
(1) Message in QSYSOPR (2) Work with problem (3) Select problem (4) Report problem (5) Contact information (6) Select severity (7) Select service provider (8) Send request now (9) Exit	OPENED				
	SENT	→	(10) DSPMSG (11) Work with problem (12) Select problem (13) Analyze problem (14) Contact information (15) User ID/Password (16) Operator present ? (17) Problem analysis	OPENED	
	SENT	←	(18) Analysis results (19) Report problem (20) Contact information (21) Select severity (22) Select service provider (23) Send request now	READY	
				Message in job log	ANSWERED

RV2N418-1

Figure 35. Operations for System-Detected Hardware Problem

Operations for detecting and correcting a hardware problem

This example explains each step shown in Figure 35.

At the Service Requester:

1. A Message arrives in the QSYSOPR message queue on the service requester:

* Equipment check on DEV DKT01

The system operator looks at the message and decides to report the problem.

2. Using the Work with Problem (WRKPRB) command, the system operator displays the problem log.
3. The system operator types an 8 (Work with problem) in the *Opt* column next to the problem ID that corresponds to the diskette failure and presses the Enter key.
4. The system operator selects option 2 (Report problem) from the Work with Problem menu and presses the Enter key.
5. The system operator types any changes necessary to the contact information on the Verify Contact Information display and presses the Enter key.
6. The system operator selects the severity of the problem from the Select Problem Severity menu and presses the Enter key.

7. The system operator selects a service provider from the Select Service Provider display and presses the Enter key.
8. The system operator selects option 1 (Send service request now) from the Select Reporting Option display and presses the Enter key to send the request.
9. The system operator presses F3 (Exit) as many times as necessary to exit the problem log.

The problem record is sent to the service provider and the network operator at the service provider is notified with a message.

At the Service Provider:

10. The network operator displays the service provider message queue defined in the service provider attributes by using the Change Service Provider Attributes (CHGSRVPVDA) command.
The name of the system on which the problem originated and the problem ID are contained in the message.
11. The network operator uses either the Work with Problem (WRKPRB) command or presses F14 (Run problem analysis) from the service provider message queue display to show the Work with Problem display.
12. The network operator types an 8 (Work with problem) in the *Opt* column next to the problem and presses the Enter key.

Note: The problem log on the service provider contains both local and remote problems.

13. The network operator selects option 1 (Analyze Problem) on the Work with Problem display and presses the Enter key.
14. The network operator presses F10 (Display contact information) to see who to contact at the service requester.
15. The network operator types the user ID and password on the Specify Sign On Information display.
16. The network operator selects option 1 (Operator present) from the Operator Intervention display if the operator is present at the service requester.
If no operator is present at the service requester, problem analysis is ended because an operator is needed to insert and remove diskettes.

Note: The next display is the first one shown from the service requester.

17. Problem Analysis displays are shown. Some displays provide instructions and other displays ask for information. The operator at the service requester follows the instructions provided to insert and remove diskettes from the diskette drive.
18. The network operator is returned to the service provider and the results of problem analysis are stored in the problem log on both systems. The results are shown on the Problem Analysis Complete display as a possible causes list. This list is a guide for taking the proper actions. The analysis results can be sent in a service request to IBM service support (*IBMSRV) if a hardware failure is the most probable cause.
The network operator may want to add a note to the problem record at this time.
 - a. The network operator presses F3 (Exit) to return to the Work with Problem menu.
 - b. The network operator types a 12 (Enter notes) next to the problem record and presses the Enter key.

- c. The network operator types the extra information that might help describe the problem.
 - d. The network operator presses F3 (Exit) to exit the Enter Notes display.
 - e. The network operator selects option 1 (Save notes and exit) from the Exit Notes display and presses the Enter key.
 - f. The network operator types an 8 (Work with problem) next to the problem record and presses the Enter key.
 - g. The network operator selects option 2 (Report problem) from the Work with Problem display.
 - h. The network operator continues with step (20).
19. The network operator presses F6 (Prepare service request) on the Problem Analysis Complete display.
 20. The network operator types any changes necessary to the contact information on the Verify Contact Information display and presses the Enter key.
 21. The network operator selects the severity of the problem from the Select Problem Severity menu and presses the Enter key.
 22. The network operator selects IBM service support (*IBMSRV) from the Select Service Provider display and presses the Enter key.
 23. The network operator selects option 1 (Send service request now) from the Select Reporting Option display to report the problem.
 24. IBM service support (*IBMSRV) analyzes the symptom string and other problem information and decides whether to send a PTF or to send a service representative to the service requester.
 25. IBM service support (*IBMSRV) sends a message to the service provider job log indicating the chosen course of action. The network operator uses the Display Job Log (DSPJOBLOG) command and presses F10 (Display detailed messages) to look at the results of the service request. The message from IBM service support (*IBMSRV) is also added to the information in the problem log.

User-detected performance problem

In this example, an OS/400 service requester is having a problem with system response time. The following specific assumptions are made for this problem situation:

- The alert status (ALRSTS) network attribute is set to *UNATTEND.
- A user makes a telephone call to report the problem to the service provider.
- The network operator at the service provider is periodically monitoring the service provider message queue and the Work with Alerts display.

Service Requester			Service Provider		Next Service Provider* System
Action	Status	↔	Action	Status	
(1) Telephone call		→	(2) Run ANZPRB (3) Another AS/400 (4) System entitled (5) Select service requester (6) User ID/Password (7) Operator present? (8) Problem analysis (9) Performance problem (10) Action plan		
(11) Create problem record	SENT	←	(11) Create problem record (12) PTF search (13) Report problem (14) Contact information (15) Select severity (16) Select service provider (17) Send request now	READY	
		→	Message in job log	SENT →	*IBMSRV (18) Analyze (19) Answer
				←	
				ANSWERED	

RV2N419-1

Figure 36. Operations for a User-Detected Performance Problem

Operations for a user-detected performance problem

This example explains each step shown in Figure 36.

At the Service Requester:

1. From the service requester, a user makes a telephone call to the network operator at the service provider to report the problem.

At the Service Provider:

2. The network operator starts problem analysis using the Analyze Problem (ANZPRB) command.
3. The network operator selects option 2 (Another AS/400) from the Select Type of System menu.
4. The network operator selects option 1 (Enrolled as a service requester) from the Another AS/400 menu.
5. While the cursor is on the Remote control point field on the Specify System Information display, the network operator presses F4 (Prompt) to select a service requester from the list of systems entitled to service.
6. The network operator types the user ID and password needed to sign on the service requester on the Specify System Information display.
7. The network operator presses the Enter key when the Attendance Warning display is shown.
8. The network operator runs problem analysis directly on the service requester.
9. The network operator selects option 2 (System performance problem) from the Analyze a New Problem menu.

10. The network operator follows the instructions provided on the System Performance Problem display.
11. Because no one is at the service requester to provide assistance, the network operator selects option 2 (Create problem record) and selects the appropriate options to describe the problem.

Note: Because this is a user-detected problem, it is especially helpful if a note is added to the problem record with additional information about the problem.
12. The network operator waits while the system searches for a PTF to correct the problem. No PTF is found.
13. The Report Problem display is shown and the network operator selects the option to prepare a service request.
14. The network operator types any changes necessary to the contact information on the Verify Contact Information display and presses the Enter key.
15. The network operator selects the severity of the problem from the Select Problem Severity menu and presses the Enter key.
16. The network operator selects IBM service support (*IBMSRV) from the Select Service Provider display and presses the Enter key.
17. The network operator selects option 1 (Send service request now) from the Select Reporting Option display to report the problem.
18. IBM service support (*IBMSRV) analyzes the symptom string and other problem information and decides whether to send a PTF or to send a service representative to the service requester.
19. IBM service support (*IBMSRV) sends a message to the service provider job log indicating the chosen course of action. The network operator uses the Display Job Log (DSPJOBLOG) command and presses F10 (Display detailed messages) to look at the results of the service request.

Application-detected field value problem

In this example, an application tests a field to determine which function to perform. The field contains an unexpected value. The following assumptions are made for this example:

- The problem is new.
- The system is attended by an operator.

The following sequence of events would happen at the service requester system:

- The application detects the problem.
- The application uses the First Failure Data Capture API to report the problem.
- The system creates a problem record.
- The operator uses the Work with Problem (WRKPRB) command to view the record.
- The operator selects option 8 to work with the problem.
- The operator selects option 2 to report the problem.

The following sequence of events would happen at the service provider system:

- OS/400 searches for a PTF with the same symptom string as the problem.
- The application developer creates a PTF. The application developer on the service provider can only create PTFs for non-IBM products.

- The operator sends the PTF (using the SNDPTF command) to the service requester.

Then, the service requester operator applies the PTF or the service provider applies the PTF using the Apply Remote PTF (APYRMTPTF) command. A change request is automatically submitted that can be viewed to determine the status of this command. You can use the Work with Submitted Change Request (WRKSBMCRQ) command to monitor the status.

Alert reported printer problem

In this example, an OS/400 service requester has a problem with a printer. The following specific assumptions are made for this problem situation:

- The alert status (ALRSTS) network attribute is set to *UNATTEND.
- A system operator is not present.
- An alert is created and is sent to the service provider.
- The network operator at the service provider is periodically monitoring the service provider message queue and the Work with Alerts display.
- The log remote problems (LOGRMTPRB) service provider attribute is set to *ALL.

Figure 37 describes the steps used to work with this problem.

Service Requester			Service Provider		Next Service Provider System
Action	Status	↔	Action	Status	
(1) Alert created (2) Alert sent		→	(3) Message in QSYSOPR (4) WRKALR (5) Recommended actions (6) Display details (alerts) (7) Work problem (option 9) (8) Work problem (option 8) (9) Analyze problem (10) User ID/Password (11) Operator present? (12) DSPMSG QSYSOPR (remote) (13) End		

RV2N420-1

Figure 37. Operations for a Printer Problem

Operations for a system-detected printer problem

This example explains each step shown in Figure 37.

At the Service Requester:

1. A message arrives in the QSYSOPR message queue on the service requester indicating a printer problem has occurred.
2. An alert is created and is sent to the service provider.

At the Service Provider:

3. A message, indicating an alert has been received, arrives in the system operator message queue (QSYSOPR).

4. The network operator types the Work with Alerts (WRKALR) command on a command line and presses the Enter key.
5. The network operator types a 5 (Display recommended actions) in the *Opt* column next to the new alert on the Work with Alerts display. The information on the display shows the causes of the failure and provides a list of actions that may correct the problem. The actions indicate the network operator should display the QSYSOPR message queue at the service requester.
6. The network operator presses F17 (Display detail) on the Display Recommended Actions display. The system message indicates a problem with print device PRT01. F3 (Exit) is pressed to return to the Work with Alerts display.
7. The network operator presses F11 (Display user/group) and then types a 9 (Work with problem) in the *Opt* column next to the alert.

Note: Because the Log remote problem service provider attribute is set to *ALL, a problem record was created as a result of this alert.

8. The network operator types an 8 (Work with problem) in the *Opt* column of the problem record shown and presses the Enter key.
9. The network operator then selects option 2 (Analyze problem) from the Work with Problem menu.
10. The network operator then types the user ID and password needed to sign on to the service requester.
11. The network operator presses the Enter key on the Attendance Warning display and proceeds with the analysis until a display with a command line is reached.
12. On a command line, the network operator types:


```
DSPMSG QSYSOPR
```

and then responds to the message that is holding the printer operations for PRT01.
13. The network operator completes analysis and ensures the results are recorded in the problem logs of both systems. A call is made to the service requester location to verify the printer is running. If the printer is running, the problem is solved; if not, the network operator can send a service request to the hardware service provider or IBM service support (*IBMSRV).

Send PTF Order (SNDPTFORD) from a service requester

In this example, the system operator at an OS/400 service requester wants to order a specific PTF, SF01728, and the corresponding cover letter. The following specific assumptions are made for this problem situation:

- The network operator at the service provider is periodically monitoring the service provider message queue.
- The log remote problems (LOGRMTPRB) service provider attribute is set to *ALL.
- The send PTFs automatically (SNDPTFAUTO) service provider attribute is set to *YES.
- The maximum distribution (MAXDST) service provider attribute is set to 500KB.
- The preferred language code for the service requester is 2923 (Dutch).
- The PTF save file is on the service provider and is released.
- The cover letter is available as a physical file member in the file QGPL/QAPZCOVER.

Figure 38 describes the steps used to send and process this request.

Service Requester			Service Provider		Next Service Provider System
Action	Status	↔	Action	Status	
(1) SNDPTFORD (2) Contact information (3) Select service provider (4) Send request now	READY PREPARED SENT ANSWERED	→ ←	(5) Create problem record (6) Check provider message (7) Send PTFs automatically	ANSWERED	

RV2N421-1

Figure 38. Operations to Send a PTF Order and Distribution

Operations for a PTF order from a service requester

This example explains each step shown in Figure 38.

At the Service Requester:

1. The system operator at the service requester types the following on a command line and presses F4 (Prompt):

```
SNDPTFORD PTF(SF01728) RMTCPNAME(*SELECT)
```

Note: More PTFs can be specified when a plus (+) sign is typed in the More values field.

2. The system operator checks the contact information on the Verify Contact Information display so that if any problems occur during the processing of the PTF order, the network operator at the service provider will know who to contact. At this point, the language code field should also be verified. For this example, the code for Dutch is 2923.

Note: Use the Work with Contact Information (WRKCNTINF) command, then select option 2 (Work with local service information) to set the default contact information.

A list of service providers is presented to allow the system operator to select the destination of the PTF order.

3. The system operator selects a service provider from the Select Service Provider display as the destination for the PTF order.
4. At the top of the Select Reporting Option display, a problem record ID is displayed. This is used to identify and work with the PTF order until the PTF is received and the problem record is closed. From this display, the system operator selects option 1 (Send service request now).

At the Service Provider:

5. The service requester establishes the connection to the service provider, and a problem record with the same problem ID as on the service requester is created in the service provider problem log.

When the PTF order is received, the service provider searches for the save file and cover letter, which are on the system and available (released) for distribution.

6. An informational message is placed on the service provider message queue to let the network operator know a PTF order has been received and processed.
 - * PTF order received from NETID.RMTCPCNAME, PTFs sent.

NETID.RMTCPCNAME is the network ID and the remote control point name for the service requester.

7. The PTF save file is sent over to the service requester automatically as a response to the PTF request.
8. The PTF and cover letter are automatically received into a save file and physical file member, respectively, at the service requester.

The network operator can press F14 (Run problem analysis) to go directly to the problem record associated with the PTF order for additional information.

When things do not work correctly

Whether System Manager runs correctly depends on the iSeries server configuration, the communications and network configuration, and the setup of the licensed program. After you have completed the setup procedures and are using the System Manager functions, one or more of the following situations can occur. The description of the situation is followed by a list of possible reasons and explanations.

Problem records not created when alerts are received

Possible reasons include:

- The log remote problems (LOGRMTPRB) service provider attribute is not set to *ALERT or *ALL.

Use the Display Service Provider Attributes (DSPSRVPVDA) command to verify the current value and the Change Service Provider Attributes (CHGSRVPVDA) command to change the value if necessary.

- The Log remote problems (LOGRMTPRB) value in the service requester information is not set to *ALERT or *ALL, or refers to the service provider attribute (*SRVPVDA), which is not set to *ALERT or *ALL.

Use the Work with Service Requesters (WRKSRVRQS) command to verify the current value in the service requester information and to change the value if necessary.

- A problem record should not have been created. Only alerts that refer to system-detected problems and have an associated problem identifier will result in problem records being created at the service provider.

Problem records not created when service requests are received

Possible reasons include:

- The log remote problems (LOGRMTPRB) service provider attribute is not set to *SRVRQS or *ALL.

Use the Display Service Provider Attributes (DSPSRVPVDA) command to verify the current value and the Change Service Provider Attributes (DSPSRVPVDA) command to change the value if necessary.

- The Log remote problems (LOGRMTPRB) value in the service requester information is not set to *SRVRQS or *ALL, or refers to the service provider attribute (*SRVPVDA), which is not set to *SRVRQS or *ALL.

Use the Work with Service Requesters (WRKSRVRQS) command to verify the current value in the service requester information, and to change the value if necessary.

Cannot find messages on the service provider

Possible reasons include:

- You may be looking for them in the wrong place.

Use the Work with Service Requesters (WRKSRVRQS) command to determine the name of the message queue for the specific service requester. If the value for the message queue is *SRVPVDA or the message queue does not exist, use the Display Service Provider Attributes (DSPSRVPVDA) command to determine the default message queue for the service provider. If this message queue does not exist, check the system operator message queue, QSYSOPR.

PTFs not sent

Possible reasons include:

- The PTF does not exist, is not released, or is associated with a product that is not supported.

Use the Display Program Temporary Fix (DSPPTF) command to check the status of the PTF on your system. If the PTF does not exist, order the PTF from your next level service provider. If the PTF is not released, use the Release Program Temporary Fix (RLSPTF) command to release it.

If the product the PTF is associated with is not supported, use the Work with Supported Products (WRKSPTPRD) command to verify the product is supported, or add support for the product if necessary.

- The send PTFs automatically (SNDPTFAUTO) service provider attribute is set to *NO.

Use the Display Service Provider Attributes (DSPSRVPVDA) command to verify the current value and the Change Service Provider Attributes (CHGSRVPVDA) command to change the value if necessary.

- The send PTFs automatically (SNDPTFAUTO) value in the service requester entry information is set to *NO.

Use the Work with Service Requesters (WRKSRVRQS) command at the service provider to verify the current value in the service requester entry information and to change the value if necessary.

- The System Manager licensed program is not started on the central site system.

Start the System Manager program using option 2 (Start System Manager) on the System Manager main menu.

- The Managed System Services licensed program is not started on the managed system or the central site system.

Start the Managed System Services program by selecting option 20 (Managed System Services) from the System Manager menu and then selecting option 2 (Start Managed System Services) from the Managed System Services main menu. You can also configure or end Managed System Services from this menu.

- If you are distributing a PTF and the distribution queue is set to *SNADS in the Work with Server Requesters (WRKSRVRQS) command, you must have *USE authority to the Send PTF (SNDPTF) command.

- If you are distributing a PTF and the distribution queue is set to *SVDS in the Work with Server Requesters (WRKSRVRQS) command, you must have *USE authority to the Copy PTF (CPYPTF) command.

PTFs sent but never arrive

Possible reasons include:

- The System Manager licensed program is not started on the central site system.
Start the System Manager program using option 2 (Start System Manager) on the System Manager main menu.
- The Managed System Services licensed program is not started on the managed system or the central site system.
Start the Managed System Services program by selecting option 20 (Managed System Services) from the System Manager menu and then selecting option 2 (Start Managed System Services) from the Managed System Services main menu. You can also configure or end Managed System Services from this menu.
- The QSNADS subsystem is not active on the service provider or the service requester.
Start the QSNADS subsystem on either system or both systems using the Start Subsystem (STRSBS) command.
- The SNADS distribution queue is held at the service provider.
Use the Work with Distribution Queue (WRKDSTQ) command to release the queue and ensure that PTFs are being distributed. You may want to use the position-to function or search function provided with the WRKDSTQ command for managing the distribution queues.
- The configuration for SNADS is not correct if the PTF was sent using object distribution.
Check the system operator (QSYSOPR) and QSRVBAS message queues on the service requester and service provider for messages relating to SNADS routing errors. Use the Configure Distribution Services (CFGDSTSRV) command to verify and change the configuration.
- The job for receiving PTFs (QESTP in the QSNADS subsystem) could not be started or has ended.
Check the system operator message queue (QSYSOPR) on the service requester for message CPZ8C45 or CPI8C45.

PTF applied by service provider does not solve the problem

Possible reasons:

- A corequisite PTF also needs to be applied.
Check the PTF cover letter for a list of prerequisites and corequisites that must also be applied.
- The PTF has special instructions that have not been performed.
Check the PTF cover letter for any special instructions you may need to perform.
- The PTF was set to be applied at the next IPL.
Use the Display PTF (DSPPTF) command to verify the status of the PTF, and use the Display Job Log (DSPJOBLOG) command to check the job log from the time the PTF was applied. The job name for delayed PTFs is SCPF. Now you can use the Apply Program Temporary Fix (APYPTF) command to apply the PTF again and then do an IPL of the system again.
- The attempt to apply a PTF has failed
Use the Display PTF (DSPPTF) command to verify the status of the PTF, and use the Display Job Log (DSPJOBLOG) command to check the job log from the time you attempted to apply the PTF. Now you can use the Apply Program Temporary Fix (APYPTF) command to apply the PTF again and then do another IPL of the system.

Takes a long time to add product support by using Work with Supported Products command

Possible reason:

- Many PTF save files for the product are already stored on the system. Information must be read from these files when a product is supported. This may require a significant amount of processing.

Cannot do remote problem analysis

Possible reasons include:

- The system value for remote sign-on QRMTSIGN on the service provider is not set to *VERIFY.

Use the Display System Value (DSPSYSVAL) command to verify the current value and the Change System Value (CHGSYSVAL) command to change the value if necessary.

- The QAUTOVRT system value on the service provider is not large enough for a virtual device to be created.

Use the Display System Value (DSPSYSVAL) command to verify the current value and the Change System Value (CHGSYSVAL) command to change the value if necessary.

- The user profile or password used was not correct.

Verify the user ID and password with the system operator at the service requester.

- The user profile for remote problem analysis is not authorized to the Work with Problem (WRKPRB), Analyze Problem (ANZPRB), and Print Error Log (PRERRLOG) commands on the service requester.

Use the Grant Object Authority (GRTOBJAUT) command to grant the authority for these commands to the user profile for running problem analysis.

Service requester is not entitled

Possible reasons include:

- The service request or PTF order was sent to the wrong system.

- No entry for the service requester exists on the service provider.

Use the Work with Service Requesters (WRKSRVRQS) command to add the service requester to the list of systems entitled to service.

- The machine type and serial number of the service requester does not exactly match that in the service requester information. This will happen if you use the copy service requester function but do not change the serial number after the copy is complete.

Use the Work with Service Requesters (WRKSRVRQS) command to change the machine type or serial number in the service requester information.

- User-based pricing information is incorrect.

The service requester is not included in the list of licenses requested. Check the license information using the Work with License Information (WRKLCINF) command. If the system or feature is not listed, select 1 (Add license key) or 2 (Change) the license key information.

Appendix D. Example product exit programs

The following is an example of three programs that can be used as exit programs to manage a language product load. The first example program (QxxEXLNG) can be both a preoperation exit program and a postoperation exit program. Because QxxEXLNG calls two other programs, QxxVFYF and QxxVFYO, all three programs must be specified as preoperation exit programs if QxxEXLNG is used as a preoperation exit program. Recall that the preoperation exit programs are specified when the product load object is created.

```
/******  
/*  
/* PROGRAM NAME: QxxEXLNG  
/*  
/*  
/* This program is a sample exit program for the language  
/* part of a product. This program is called  
/* BEFORE and AFTER the language objects  
/* are saved, restored, deleted, or checked.  
/*  
/* Exceptions signaled: CPF3D95 - Exit program processing not  
/* complete.  
/* CPF3D98 - Exit program processing found  
/* error in product.  
/* (ONLY for *SAVLNG and *CHKLNG)  
/*  
/* Exceptions monitored: CPF0000  
/* CPF8A16 - Document objects not deleted.  
/* &1; objects deleted.  
/*  
/* Unless otherwise noted, secondary language libraries are  
/* handled like other libraries. If you need to special case  
/* your code for secondary language libraries, code will need  
/* to be added.  
/*  
/* The parameters for this program are:  
/*  
/* 1 CHAR(10) - Function to be performed  
/* *RSTLNG - restore language  
/* *SAVLNG - save language parts  
/* *DLTLNG - delete language parts  
/* *CHKLNG - check language parts  
/* (used by CHKPRDOPT for *AFTER)  
/*  
/* 2 CHAR(10) - Program called *BEFORE or *AFTER  
/*  
/* 3 CHAR(4) - Language ID (eg. 2924)  
/*  
/* 4 CHAR(10) - Exit program library  
/*  
/* 5 CHAR(10) - Object library  
/*  
/* 6 CHAR(10) - Create library  
/*  
/* 7 CHAR(10) - Restore library  
/*  
/* 8 CHAR(12) - Object folder  
/*  
/* 9 CHAR(12) - Create folder  
/*  
/* 10 CHAR(12) - Restore folder  
/*
```

```

/*      11 CHAR(6)      - Current Version/Release/Modification level */
/*
/*      12 CHAR(6)      - Restore Version/Release/Modification level */
/*
/*      13 CHAR(9980) - Array of missing library objects          */
/*                      CHAR(10) Object name CHAR(10) Object type */
/*                      NOTE: The maximum character string length */
/*                      that can be passed is 9999. So 499        */
/*                      library objects 20 characters in         */
/*                      length can be passed. (20*499 = 9980)*/
/*
/*      14 CHAR(4)      - Number of missing library objects       */
/*                      (Maximum number of 499)                   */
/*
/*      15 CHAR(6300) - Array of missing folders                  */
/*                      CHAR(63) Folder name                       */
/*                      NOTE: The maximum character string length */
/*                      that can be passed is 9999. So 100       */
/*                      folders 63 characters in length can      */
/*                      be passed. (63*100 = 6300)                */
/*
/*      16 CHAR(4)      - Number of missing folders               */
/*                      (Maximum number of 100)                   */
/*
/*****
QSYS/PGM      PARM(&funct &pgmcall &lngid &exitlib &objlib +
              &crtlib &rstlib &objflr &crtflr &rstflr +
              &curvrm &rstvrm &misobj &numobj +
              &misflr &numflr);

/*****
/*
/*      Declare parameter variables.
/*
/*****
QSYS/DCL &funct  TYPE(*char) LEN(10) /* Requested function          */
QSYS/DCL &pgmcall TYPE(*char) LEN(10) /* Program called before/after*/
QSYS/DCL &lngid  TYPE(*char) LEN(4)  /* Language ID (eg. 2924)    */
QSYS/DCL &exitlib TYPE(*char) LEN(10) /* Lib containing exit pgms  */
QSYS/DCL &objlib  TYPE(*char) LEN(10) /* Library containing product */
QSYS/DCL &crtlib  TYPE(*char) LEN(10) /* Library when product crtd  */
QSYS/DCL &objflr  TYPE(*char) LEN(12) /* Folder containing product  */
QSYS/DCL &rstflr  TYPE(*char) LEN(12) /* Folder for prod after rst  */
QSYS/DCL &curvrm  TYPE(*char) LEN(6)  /* Current Ver/Rls/Mod level  */
QSYS/DCL &misobj  TYPE(*char) LEN(9980) /* List of missing objects   */
QSYS/DCL &numobj  TYPE(*char) LEN(4)   /* Number of missing objects*/
/*                      in binary      */
QSYS/DCL &misflr  TYPE(*char) LEN(6300) /* List of missing folders   */
QSYS/DCL &numflr  TYPE(*char) LEN(4)   /* Number of missing folders*/
/*                      in binary      */

/*****
/*
/*      Declare local work variables
/*
/*****
QSYS/DCL &rtntvalue TYPE(*char) VALUE(' ') /* Return value          */
QSYS/DCL &errsent  TYPE(*dec)  VALUE(0) /* Error sent return value */
QSYS/DCL &misobj#  TYPE(*dec)  LEN(4)   /* Number of missing objects*/
/*                      in decimal      */
QSYS/DCL &misflr#  TYPE(*dec)  LEN(4)   /* Number of missing folders*/
/*                      in decimal      */

/*****
/*
/*      Unexpected errors are handled by sending an escape message to */

```



```

/* the calling program. */
/* (NOTE: MONMSG must precede any commands.) */
/* */
/*****/
QSYS/MONMSG MSGID(CPF0000) +
      EXEC(QSYS/GOTO errexit)

/*****/
/* */
/* Change number of missing objects/folders from binary to decimal*/
/* */
/*****/
QSYS/CHGVAR      &misobj#  %BINARY(&numobj)
QSYS/CHGVAR      &misflr#  %BINARY(&numflr)

/*****/
/*          B E F O R E          */
/* */
/* Do the appropriate operations for the requested function. */
/* Note that no operation is performed if the requested */
/* function is not one that this program expects to handle and */
/* no messages are sent: */
/* */
/* -- For restoring the language objects (*RSTLNG) into the */
/* specified restore library or folder, */
/* perform all the necessary install setup. */
/* */
/* -- For saving the language objects (*SAVLNG), call a program */
/* (either QxxVFYO or QxxVFYF) to verify the list of missing */
/* objects and/or folders. */
/* */
/* -- For deleting the language objects (*DLTLNG), */
/* perform all necessary product cleanup. */
/* */
/*****/
/* For a save operation, this module calls a program to verify */
/* that the objects or folders that are missing are OK to be */
/* missing for the save to continue. If an object or folder is */
/* missing and shouldn't be, an error message is sent and the */
/* save is stopped. */
/* */
/*****/
/* If program to run BEFORE */
QSYS/IF          COND(&pgmcall = '*BEFORE') +
      THEN(QSYS/DO)          /* Check what function to perform*/

/*****/
/* If language restore, call the programs to restore library objects*/
/* and folders. (In pre-restore processing, the exit programs are */
/* found in QTEMP.) */
/* */
/*****/
      QSYS/IF          COND(&funct = '*RSTLNG') +
      THEN(QSYS/DO)

/*****/
/* */
/* This example assumes the product contains some database */
/* file objects that must be deleted to restore new versions. */
/* Note that this program is called to handle BOTH install */
/* into the product library, and to install into a secondary */
/* language library. The &rstlib parameter indicates what */
/* library is to contain the objects. */
/* */
/* Delete all objects (normally only database files) that */
/* must be removed from the system in order to assure new */
/* objects can be successfully restored. The "error" of an

```

```

/* object not found is ignored (other than to remove the      */
/* CPF2105 escape message from the program message queue), to */
/* handle the first installation of the product.              */
/*                                                            */
/*****

      QSYS/DLTF      FILE(&rstlib/xxxxyyyy)
      QSYS/MONMSG   MSGID(cpf2105) +
      EXEC(QSYS/RCVMSG PGMQ(*same) MSGTYPE(*excp) +
      RMV(*yes))

/*****
/*
/* Before restoring the language part of the product,
/* the language folder is deleted. (eg. PRODFLR/MRIxxxx)
/* The RSTDLO command uses ALWOBJDIF(*NONE) to restore the
/* folder.
/*
/* Note that this program is called to handle BOTH the install
/* into the primary language folder and the install into a
/* secondary language folder.
/*
/*****

      /*****
      /* Delete language folder if it exists      */
      /*****
      QSYS/DLTDLO DLO(*ALL) FLR(&rstflr *CAT '/' *CAT 'MRI' *CAT &lngid)
      QSYS/MONMSG MSGID(CPF8A16) +
      EXEC(QSYS/DO)
      QSYS/RCVMSG PGMQ(*SAME) MSGTYPE(*EXCP) RMV(*YES)
      QSYS/RCVMSG PGMQ(*SAME) MSGTYPE(*DIAG) RMV(*NO) +
      KEYVAR(&key) MSGID(&id)
      QSYS/IF COND(&ID = 'CPF8A77') +
      THEN(QSYS/RMVMSG PGMQ(*SAME) MSGKEY(&key))
      QSYS/ELSE (QSYS/GOTO errexit)
      QSYS/ENDDO
      QSYS/ENDDO      /* *RSTLNG complete - exit      */
      QSYS/ELSE QSYS/DO      /* not *RSTLNG, check if *SAVLNG */

/*****
/* If language save, call the programs to verify missing objects
/* and folders.
/*
/*****
      QSYS/IF      COND(&funct = '*SAVLNG') +
      THEN(QSYS/DO)
      /* Verify list of missing objects*/

      QSYS/IF      COND(&misobj# %= 0) +
      THEN(QSYS/DO)
      QSYS/CALL PGM(&exitlib/QXXVFO) +
      PARM(&lngid &objlib &curvrn &misobj +
      &misobj# &errsent &rtnvalue)

      /* If error occurred, exit      */
      QSYS/IF      COND(&rtnvalue = 'Q') +
      THEN(QSYS/GOTO errexit)
      QSYS/ENDDO

      /* Verify list of missing folders*/
      QSYS/IF      COND(&misflr# %= 0) +
      THEN(QSYS/DO)
      QSYS/CALL PGM(&exitlib/QXXVFF) +
      PARM(&lngid &objflr &curvrn &misflr +
      &misflr#; &errsent &rtnvalue)

      /* If error occurred, exit      */

```

```

        QSYS/IF      COND(&rtnvalue = 'Q') +
        THEN(QSYS/GOTO errexit)
    QSYS/ENDDO

        /* If verify error sent, send +
        escape message CPF3D98.      */
    QSYS/IF      COND(&errsnt = 1) +
    THEN(QSYS/DO)
        QSYS/SNDPGMMSG MSGID(CPF3D98) +
        MSGF(QCPFMSG) +
        MSGTYPE(*ESCAPE)

    QSYS/ENDDO
    QSYS/ELSE QSYS/DO
    QSYS/ENDDO

    QSYS/ENDDO          /* *SAVLNG complete - exit      */

    QSYS/ELSE QSYS/DO   /* not *SAVLNG, check if *DLTLNG */
/*****
/* If language delete, ...
/*****
        QSYS/IF      COND(&funct = '*DLTLNG') +
        THEN(QSYS/DO)
        QSYS/ENDDO          /* *DLTLNG complete - exit      */

        QSYS/ELSE QSYS/DO /* Not one of the supported      */
        QSYS/GOTO errexit
        QSYS/ENDDO          /* functions - issue error      */

/*****
/* *BEFORE processing complete - exit QxxEXLNG
/*****
        QSYS/ENDDO          /* Not *SAVLNG      */
        QSYS/ENDDO          /* Not *RSTLNG      */
        QSYS/ENDDO          /* End *BEFORE processing      */
        QSYS/ELSE QSYS/DO   /* Not *BEFORE, check if *AFTER */

/*****
/*
/*
/* Do the appropriate operations for the requested function.
/* Note that no operation is performed if the requested
/* function is not one that this program expects to handle:
/*
/* -- For restoring the language objects (*RSTLNG) into the
/* product library, perform any
/* functions required of the product following the install.
/*
/* -- For saving the language objects (*SAVLNG) from the
/* product library, perform any
/* functions required of the product following the save.
/*
/* -- For checking the language objects (*CHKLNG), verify that
/* missing library objects and folders are OK to be missing.
/*
/*****
        /* If program to run AFTER, check +
        what function to perform      */
    QSYS/IF      COND(&pgmcall = '*AFTER') +
    THEN(QSYS/DO)
        QSYS/IF      COND(&funct = '*RSTLNG') +
/*****
/* AFTER restore processing.
/*****
    QSYS/ENDDO          /* End *RSTLNG - exit      */

```

```

        QSYS/ELSE QSYS/DO      /* Not *RSTLNG, check if *SAVLNG */

        QSYS/IF              COND(&funct = '*SAVLNG') +
        THEN(QSYS/DO)
        QSYS/ENDDO
        QSYS/ELSE QSYS/DO    /* Not *SAVLNG, check if *CHKLNG */

/*****
/* If language check, call the programs to verify missing objects */
/* and folders. */
/*****
        QSYS/IF              COND(&funct = '*CHKLNG') +
        THEN(QSYS/DO)
                                /* Verify list of missing objects*/

        QSYS/IF              COND(&misobj# %= 0) +
        THEN(QSYS/DO)
                QSYS/CALL PGM(&exitlib/QXXVFO) +
                PARM(&lngid &objlib +
                    &curvrm &misobj +
                    &misobj#; &rtvalue)

        QSYS/ENDDO

                                /* Verify list of missing folders*/
        QSYS/IF              COND(&misflr# %= 0) +
        THEN(QSYS/DO)
                QSYS/CALL PGM(&exitlib/QXXVFF) +
                PARM(&lngid &objflr +
                    &curvrm &misflr +
                    &misflr# &rtvalue)

        QSYS/ENDDO

                                /* If verify error occurred, exit*/
        QSYS/IF              COND(&rtvalue = 'Q') +
        THEN(QSYS/DO)
                QSYS/SNDPGMMSG MSGID(CPF3D98) +
                MSGF(QCPFMSG) +
                MSGTYPE(*ESCAPE)

        QSYS/ENDDO

        QSYS/ENDDO          /* *CHKLNG complete - exit */

        QSYS/ELSE QSYS/DO /* Not a supported function */
        QSYS/GOTO errexit
        QSYS/ENDDO          /* issue error message */

/*****
/* *AFTER processing complete - exit QxxEXLNG */
/*****
        QSYS/ENDDO          /* Not *SAVLNG */
        QSYS/ENDDO          /* Not *RSTLNG */
        QSYS/ENDDO          /* End *AFTER processing */

        QSYS/ELSE QSYS/DO /* Not a supported program call */
        QSYS/GOTO errexit
        QSYS/ENDDO          /* issue error message */

        QSYS/ENDDO          /* No more program calls to check*/

/*****
/* Normal return */
/*****
        QSYS/RETURN

/*****
/* Send error message if error occurred. */
/*****

```

```

errexit:
  QSYS/SNDPGMMSG  MSGID(CPF3D95) +
                  MSGF(QCPFMSG) +
                  MSGTYPE(*ESCAPE)
QSYS/ENDPGM

```

The following is an example of a program that processes a list of missing objects and sends a message for each object that is missing. This program is called by program QxxEXLNG.

```

/*****/
/*
/* PROGRAM NAME: QxxVFYO
/*
/* Program QxxVFYO is a sample program to verify the missing
/* library objects on a save operation. This program is called
/* by the exit program for the language part of the product
/* (QxxEXLNG), when a save operation (*SAVLNG) or a
/* check operation (*CHKLNG) are performed.
/*
/* Messages sent:      CPD3DC8 - Object missing in library.
/* Messages monitored: CPF0000
/*
/* The parameters for QxxVFYO are:
/*
/*      1 CHAR(4)  - Language ID (eg. 2924)
/*      2 CHAR(10) - Object library
/*      3 CHAR(6)  - Current Version/Release/Modification level
/*      4 CHAR(9980) - Array of missing library objects
/*                  CHAR(10) Object name
/*                  CHAR(10) Object type
/*      5 DEC(4)   - Number of missing library objects
/*      6 DEC(1)   - Error sent indicator
/*      7 CHAR(1)  - Return value (Q = error)
/*
/*****/
QSYS/PGM          PARM(&lngid &objlib &curvrm &misobj &misobj# +
                    &errsent &rtnvalue)

/*****/
/*
/* Declare parameter variables.
/*
/*****/
QSYS/DCL &lngid   TYPE(*char) LEN(4)   /* Language ID
QSYS/DCL &objlib  TYPE(*char) LEN(10)  /* Object library
QSYS/DCL &curvrm  TYPE(*char) LEN(6)   /* Cur version/release/mod
QSYS/DCL &misobj  TYPE(*char) LEN(9980) /* List of missing objects
QSYS/DCL &misobj# TYPE(*dec)  LEN(4)   /* Number of missing object
QSYS/DCL &errsent TYPE(*dec)  LEN(1)   /* Error sent indicator
QSYS/DCL &rtnvalue TYPE(*char) LEN(1)  /* Return value (Q=error)

/*****/
/*
/* Declare local work variables
/*
/*****/
/* Pointer to name in missing object list*/
QSYS/DCL &ptr1   TYPE(*dec) LEN(4) VALUE(1)
/* Pointer to name in the OK list
QSYS/DCL &ptr2   TYPE(*dec) LEN(4) VALUE(1)
/* Pointer to object type in missing list*/
QSYS/DCL &type   TYPE(*dec) LEN(4) VALUE(1)
/* Counter of items in missing obj list
QSYS/DCL &ctr1   TYPE(*dec) LEN(4) VALUE(1)
/* Counter of items in the OK list
QSYS/DCL &ctr2   TYPE(*dec) LEN(4) VALUE(1)
/* Name of missing object

```

```

QSYS/DCL &objname TYPE(*char) LEN(63)
/* Type of missing object */
QSYS/DCL &objtype TYPE(*char) LEN(10)

/*****/
/*
/* Declare your list of missing objects that are ok to be missing */
/* and the number of items in the list. The objects must be in */
/* the following format: obj name (10 char) obj type (10 char) */
/* up to a maximum of 499 objects. */
/*****/
QSYS/DCL &okobj# TYPE(*dec) LEN(4 0) VALUE(3)
QSYS/DCL &okobj TYPE(*char) +
          VALUE('FILE1      FILE      +
                pgm1      PGM      +
                msg4567890 MSGF    ')

/*****/
/*
/* Unexpected errors are handled by setting an error return value */
/* to terminate the operation. */
/*
/*
/*****/
QSYS/MONMSG MSGID(CPF0000) +
          EXEC(QSYS/GOTO errexit)

/*****/
/*
/*          V E R I F Y */
/*
/* Determine if there is a list of objects OK to be missing. */
/* If not, send an error message (CPD3DC8) for each object */
/* missing in the list. */
/*
/*
/*****/
QSYS/IF COND(&okobj# = 0) +
          THEN(QSYS/DO)

          /*****/
          /* If there is no list of objects, isolate the name and */
          /* type of each object in the missing list for error */
          /* message CPD3DC8, where */
          /*      n1xxxx = your product identifier */
          /*      n      = the option number */
          /*****/
senderr:
QSYS/CHGVAR &objname %SST(&misobj &ptr1 10)
QSYS/CHGVAR &type    (&ptr1 + 10)
QSYS/CHGVAR &objtype %SST(&misobj &type 10)

QSYS/SNDPGMMSG MSGID(CPD3DC8) +
              MSGF(QCPFMSG) +
              MSGDTA('n1xxxx' *CAT 'n      ' *CAT +
                    &lngid *CAT &curvrm *CAT +
                    &objtype *CAT &objname *CAT &objlib) +
              MSGTYPE(*DIAG)

QSYS/CHGVAR &errsent 1 /* Indicate error has been sent */

/*****/
/* Increment counter of missing objects. If there are */
/* more, point to next object in the missing list and */
/* repeat process for sending the error. */
/*
/*****/
QSYS/CHGVAR &ctr1 (&ctr1 + 1)
QSYS/IF COND(&ctr1 &lt;= &misobj#) +
          THEN(QSYS/DO)
          QSYS/CHGVAR &ptr1 (&ptr1 + 20)

```

```

        QSYS/GOTO senderr
    QSYS/ENDDO

```

```

    QSYS/ENDDO                                /* End processing when no list */
/*****
/*
/* There is a list of objects that are OK to be missing. Compare */
/* each object in the missing list with each object in the OK   */
/* list. (The pointers and counters for each list were set to   */
/* one in the declare (DCL) statements.                          */
/*
/*
/*****
    QSYS/ELSE CMD(QSYS/DO)
    cmpobj:

    QSYS/IF COND(%SST(&misobj &ptr1 20) = %SST(&okobj &ptr2 20)) +
    THEN(QSYS/DO)
    QSYS/ENDDO                                /* The objects match */

    QSYS/ELSE CMD(QSYS/DO)                    /* The objects don't match */

/*****
/* Increment counter of OK objects. If there are more */
/* objects in the OK list, point to the next object in */
/* the OK list and repeat the compare process.          */
/*****
    QSYS/CHGVAR &ctr2 (&ctr2 + 1)
    QSYS/IF COND(&ctr2 &lt;= &okobj#) +
    THEN(QSYS/DO)
        QSYS/CHGVAR &ptr2 (&ptr2 + 20)
        QSYS/GOTO CMPOBJ
    QSYS/ENDDO

/*****
/* If there are no more objects, isolate the name and */
/* type of the object for error message CPD3DC8, where */
/* n1xxxxx = your product number                      */
/* n = the option number                               */
/*****
    QSYS/ELSE CMD(QSYS/DO)
        QSYS/CHGVAR &objname %SST(&misobj &ptr1 10)
        QSYS/CHGVAR &type (&ptr1 + 10)
        QSYS/CHGVAR &objtype %SST(&misobj &type 10)

        QSYS/SNDPGMSG +
            MSGID(CPD3DC8) +
            MSGF(QCPFMSG) +
            MSGDTA('n1xxxxx' *CAT 'n ' *CAT +
                &lngid *CAT &curvrm *CAT +
                &objtype *CAT &objname *CAT &objlib) +
            MSGTYPE(*DIAG)

        QSYS/CHGVAR &errsent 1 /* Indicate error was sent */
    QSYS/ENDDO

    QSYS/ENDDO                                /* End missing object not found */
/*****
/* Increment counter of missing objects. If there are */
/* more objects in the missing list, point to the next */
/* object in the missing list and point to the first */
/* object in the OK list. Repeat the compare process. */
/*****
    QSYS/CHGVAR &ctr1 &ctr1 + 1)
    QSYS/IF COND(&ctr1 &lt;= &misobj#) +
    THEN(QSYS/DO)
        QSYS/CHGVAR &ptr1 (&ptr1 + 20)

```

```

        QSYS/CHGVAR &ptr2 1
        QSYS/CHGVAR &ctr2 1
        QSYS/GOTO cmpobj
    QSYS/ENDDO

    QSYS/ENDDO                                /* End processing when list found*/

/*****
/*
/* Normal return.
/*
/*
/*****
    QSYS/RETURN

/*****
/*
/* The following code reports unexpected errors to the calling
/* program. It sets a return value that forces termination of the
/* operation, and returns to the caller.
/*
/*
/*****
errexit:
    QSYS/CHGVAR &rtvalue 'Q'

    QSYS/ENDPGM

```

The following is an example of a program that processes a list of missing folders and sends a message for each folder that is missing. This program is called by program QxxEXLNG.

```

/*****
/*
/* PROGRAM NAME: QxxVFYF
/*
/* Program QxxVFYF is a sample program to verify the missing
/* folders on a save operation. This program is called by the
/* exit program for the language part of the product (QxxEXLNG),
/* when the LP component calls the program for a save operation
/* (*SAVLNG) or when the SZ component invokes the program for a
/* check operation (*CHKLNG).
/*
/* Exceptions signaled: CPF3DE7 - Folder xxxxxx not found.
/* Exceptions monitored: CPF0000
/*
/* The parameters for QxxVFYF are:
/*
/* 1 CHAR(4) - Language ID (eg. 2924)
/* 2 CHAR(12) - Object Folder
/* 3 CHAR(6) - Current Version/Release/Modification level
/* 4 CHAR(6300) - Array of missing folders
/* CHAR(63) - folder name
/* 5 DEC(4) - Number of missing folders
/* 6 DEC(1) - Error sent indicator
/* 7 CHAR(1) - Return value (Q=error)
/*
/*****
QSYS/PGM PARM(&lngid &objflr &curvrm &misflr &misflr#; +
&errsent &rtvalue);

/*****
/*
/* Declare parameter variables.
/*
/*****
QSYS/DCL &lngid TYPE(*char) LEN(4) /* Language ID
QSYS/DCL &objflr TYPE(*char) LEN(12) /* Object folder
QSYS/DCL &curvrm TYPE(*char) LEN(6) /* Current version/r1s/mod

```



```

QSYS/DCL &misflr  TYPE(*char) LEN(6300) /* List of missing folders */
QSYS/DCL &misflr# TYPE(*dec)  LEN(4)   /* Number of missing folder*/
QSYS/DCL &errsent TYPE(*dec)  LEN(1)   /* Error sent indicator   */
QSYS/DCL &rtvalue TYPE(*char) LEN(1)   /* Return value (Q = error)*/
/*****/
/*
/*   Declare local work variables
/*
/*
/*****/
/* Pointer to name in missing folder list*/
QSYS/DCL &ptr1   TYPE(*dec) LEN(4) VALUE(1)
/* Pointer to name in the OK list      */
QSYS/DCL &ptr2   TYPE(*dec) LEN(4) VALUE(1)
/* Counter of items in missing flr list */
QSYS/DCL &ctr1   TYPE(*dec) LEN(4) VALUE(1)
/* Counter of items in the OK list      */
QSYS/DCL &ctr2   TYPE(*dec) LEN(4) VALUE(1)
&flrname TYPE(*char) LEN(63)

/*****/
/*
/*   Declare your list of missing folders that are OK to be missing */
/*   and the number of items in the list. Each folder entry is     */
/*   63 characters. Up to a maximum of 158 folders may be         */
/*   specified (CL restriction of character string maximum of 9999).*/
/*****/
QSYS/DCL &okflr# TYPE(*dec) LEN(4) VALUE(0)
QSYS/DCL &okflr  TYPE(*char) +
VALUE('FOLDERA                                     +
      A/B/C                                          ')

/*****/
/*
/*   Unexpected errors are handled by setting an error return value */
/*   to terminate the operation.
/*
/*
/*****/
QSYS/MONMSG MSGID(CPF0000) +
EXEC(QSYS/GOTO errexit)

/*****/
/*
/*           V E R I F Y
/*
/*
/*   Determine if there is a list of folders that are OK to be
/*   missing. If not, send an error message (CPD3DE7) for each
/*   folder in the missing list.
/*
/*
/*****/
QSYS/IF      COND(&okflr#; = 0) +
THEN(QSYS/DO) /* Begin process if no list */
/*****/
/* There is no list of folders, isolate the folder name */
/* for each folder in the missing list for error message*/
/* CPD3DE7, where
/*     1MYPROD = your product number
/*     n       = the option number
/*****/
senderr:
QSYS/CHGVAR &flrname %SST(&misflr &ptr1 63)

QSYS/SNDPGMSG MSGID(CPD3DE7) +
MSGF(QCPFMSG) +
MSGDTA('1MYPROD' *CAT 'n ' *CAT &lngid *CAT +
      &curvrm *CAT '*FLR ' *CAT +
      &flrname *CAT 'QDOC ') +
MSGTYPE(*DIAG)

```

```

        QSYS/CHGVAR &errsent 1      /* Indicate error was sent */

        /*****
        /* Increment counter of missing objects. If there are */
        /* more, point to next object in the missing list and */
        /* repeat process for sending the error. */
        /*****
        QSYS/CHGVAR &ctr1 (&ctr1 + 1)
        QSYS/IF COND(&ctr1 &lt;= &misflr#) +
            THEN(QSYS/DO)
                QSYS/CHGVAR &ptr1 (&ptr1 + 63)
                QSYS/GOTO senderr
        QSYS/ENDDO

    QSYS/ENDDO                                /* End processing when no list */

/*****
/*
/* There is a list of folders that is OK to be missing. Compare */
/* each folder in the missing list with each folder in the OK */
/* list. (The pointers and counters for each list were set to */
/* one in the declare (DCL) statements.) */
/*
/*****
        QSYS/ELSE CMD(QSYS/DO)                /* Begin process when list found */

cmpflr:
        QSYS/IF COND(%SST(&misflr &ptr1 63) = %SST(&okflr &ptr2 63)) +
            THEN(QSYS/DO)
                QSYS/ENDDO                                /* The folders match */

        QSYS/ELSE CMD(QSYS/DO)                /* The folders don't match */
            /*****
            /* Increment counter of OK folders. If there are more */
            /* folders in the OK list, point to the next folder in */
            /* the OK list and repeat the compare process. */
            /*****
                QSYS/CHGVAR &ctr2 (&ctr2 + 1)
                QSYS/IF COND(&ctr2 &lt;= &okflr#) +
                    THEN(QSYS/DO)
                        QSYS/CHGVAR &ptr2 (&ptr2 + 63)
                        QSYS/GOTO cmpflr
                QSYS/ENDDO

            /*****
            /* There are no more folders, isolate the folder name */
            /* for error message CPD3DE7, where */
            /* 1MYPROD = your product number */
            /* n = the option number */
            /*****
                QSYS/ELSE CMD(QSYS/DO)
                    QSYS/CHGVAR &flrname %SST(&misflr &ptr1 63)

                    QSYS/SNDPGMMSG +
                        MSGID(CPD3DE7) +
                        MSGF(QCPFMSG) +
                        MSGDTA('1MYPROD' *CAT 'n ' *CAT &lngid *CAT +
                            &curvrm *CAT '*FLR ' *CAT +
                            &flrname *CAT 'QDOC ') +
                        MSGTYPE(*DIAG)

                    QSYS/CHGVAR &errsent 1 /* Indicate error was sent */
                QSYS/ENDDO

        QSYS/ENDDO                                /* End missing folder not found */

/*****

```

```

/* Increment counter of missing folders. If there are */
/* more folders in the missing list, point to the next */
/* folder in the missing list and point to the first */
/* folder in the OK list. Repeat the compare process. */
/*****/

QSYS/CHGVAR &ctr1 (&ctr1 + 1)
QSYS/IF COND(&ctr1 &lt;= &misflr#) +
  THEN(QSYS/DO)
    QSYS/CHGVAR &ptr1 (&ptr1 + 63)
    QSYS/CHGVAR &ptr2 1
    QSYS/CHGVAR &ctr2 1
    QSYS/GOTO cmpflr
  QSYS/ENDDO

QSYS/ENDDO /* End processing when list found*/
/*****/
/* Normal return */
/* */
/*****/
QSYS/RETURN

/*****/
/* The following code reports unexpected errors to the calling */
/* program. It sets a return value that forces termination of */
/* the operation, and returns to the caller */
/* */
/*****/
errexit:
  QSYS/CHGVAR &rtnvalue 'Q'

QSYS/ENDPGM

```


Appendix E. System Manager APIs and exit programs

The following table defines the APIs and the functions they perform:

Table 9. Change Request Manager APIs

Program	Pseudonym	Description
QNSADDCM	Add change management activity	Adds an activity to a change request description to perform a change management function. When the change request containing the added activity is submitted, the activity is to perform the specified function.
QNSSNDCM	Send Change Request Message API	Allows an application program to add messages to a change request activity that are intended to provide intermediate and final status information for the activity.
QNSUPDST	Update Status API	Instructs the Change Request Manager to update the status of the specified activity based on the information passed to it.
Exit program	Add Activity Exit Program	Each application that registers with the system can optionally provide an add activity exit program or command. The system calls the exit program or command when you select the option to add an activity to the change request description. The add activity program must follow the usage rules and accept the parameters defined by this interface.
Exit program	Change Activity Exit Program	Each application that registers with the system can optionally provide a change activity exit program or command. The system calls the exit program or command when you select the option to change an activity. The change activity program must follow the usage rules and accept the parameters defined by this interface.
Exit program	Display Activity Exit Program	Each application that registers with the Change Request Manager can optionally provide a display activity exit program or command. The system calls the exit program or command when you select the option to display or print type-specific activity information. The display activity program must follow the usage rules and accept the parameters defined by this interface.
Exit program	Run Activity Exit Program	Each application that registers with the Change Request Manager must provide a run activity exit program. The system calls the exit program or command to start activities specific to that application. The run activity program must follow the usage rules and accept the parameters defined by this interface.

These APIs are intended to support high-level languages on an iSeries server used by both IBM programs and user programs. For examples and explanations about how to use the job log to diagnose API errors, see the API topic under the Programming category in the Information Center.

Note: When you use an exit program, do not change the definition of the exit points.

Change Request APIs

The following APIs are defined by the System Manager program.

System Manager programs call the exit programs specifying the required parameters to perform the desired action. The exit program name is defined by the application.

Each program that is called has a fixed set of parameters defined by System Manager. The exit (application) program must accept those parameters.

Add Change Management Activity (QNSADDCM) API

Parameters

Required Parameter Group:

1	Change request description	Input	CHAR(20)
2	Activity	Input	CHAR(10)
3	Action value list	Input	CHAR(*)
4	Action value list length	Input	BINARY(4)
5	Destination format	Input	CHAR(10)
6	Destinations	Input	CHAR(*)
7	Number of destinations	Input	BINARY(4)
8	Condition list	Input	ARRAY OF CHAR(32)
9	Number of conditions	Input	BINARY(4)
10	Start time	Input	CHAR(40)
11	Hold	Input	CHAR(10)
12	Replace	Input	CHAR(10)
13	Text	Input	CHAR(50)
14	Error code	I/O	CHAR(*)
15	Activity added	Output	CHAR(10)

Optional Parameter:

15	Activity added	Output	CHAR(10)
----	----------------	--------	----------

The Add Change Management Activity (QNSADDCM) API adds an activity to a change request description to perform software management and data distribution. When the change request containing the added activity is submitted, the activity is run to perform the specified function.

This API can be used to perform software distribution and change management operations that are not supported by the existing System Manager commands. For example, the installation of PC software can be done using this API. This is

because the existing commands only support installation of products packaged using the System Manager product packaging support or installation of OS/400 PTFs.

Notes:

1. To apply PTFs, the actions install (*INS) or send and install (*SNDINS) should be used.
2. The Add Change Request Activity (ADDCRQA) and Change Change Request Activity (CHGCRQA) commands can be used to add or replace change request activities to perform software distribution and change management operations on OS/2[®] clients.

Authorities and Locks

Change Request Description Authority

*CHANGE

Library Authority

*EXECUTE

Change Request Description Lock

*EXCLRD

Required Parameter Group

Change request description

INPUT; CHAR(20)

The name of the change request description to which the activity is to be added. The first 10 characters specify the object name, and the second 10 characters specify the library. You can use these special values for the library name:

***CURLIB**

The current library is used to locate the object. If there is no current library, QGPL (general purpose library) is used.

***LIBL** The library list is used to locate the object.

Activity

INPUT; CHAR(10)

The name of the activity to add to the change request description. The possible values are:

***GEN** Generates an activity name of the form QACTxxxxxx where xxxxxx is a multiple of ten not already used.

***LAST**

The activity is the last to run in the change request. No other conditions may be specified.

activity-name

A user-defined activity name.

Action value list

INPUT; CHAR(*)

Specifies the required and optional parameter values needed to do the action. See the "Action Value List" on page 316 for details about the format of this parameter.

QNSADDCM API

Action value list length

INPUT; BINARY(31)

Specifies the length of the action value list in bytes. The maximum value is 12288 bytes.

Destination format

INPUT; CHAR(10)

Specifies whether the destination is a single node or a node list. The possible values are:

*NODL

Node list object name

*SNALST

List of node names

Destinations

INPUT; CHAR(*)

The target destination of the activity. The format is based on the destination format parameter. When destination format is *SNALST, the format of the destinations are an array of CHAR(20) node names:

Network identifier

CHAR(8)

Control point name

CHAR(8)

Reserved

CHAR(4)

*NETATR may be used for the network ID to indicate that the destination is in the same network as this system.

When the destination format is *NODL, the destination format is:

Node list name

CHAR(10)

Library

CHAR(10)

*LIBL and *CURLIB may be specified for the library name.

Number of destinations

INPUT; Binary(4)

Number of destinations (1 through 50) specified. If the destination format parameter is *NODL, then the number must be a 1.

Note: When the number of condition elements is 0, the activity specified is not dependent on any other activities to complete before it can start and the condition list parameter is ignored.

Condition list

INPUT; Array of CHAR(32)

The list of conditions that must be met before the activity can be run. The format for each element is:

Activity name

CHAR(10)

The activity name can be:

activity name

The activity is conditioned on the activity name specified.

generic name

The activity is conditioned on all the activities that match the generic name (partial activity followed by an asterisk (*)).

***PRV** The activity is conditioned on the activity that precedes it alphabetically.**Relation**

CHAR(3)

Valid values are *EQ, *NE, *GT, *LT, *GE, and *LE.

Reserved

CHAR(3)

Code BINARY(31)

The end code value to base conditioning on. Possible values are:

0-99 - user-specified

-1 (*SUCCESS) - end code is any value from 0-9. If specified, relation must be *EQ or *NE.

-2 (*FAIL) - end code is any value from 10-89. If specified, relation must be *EQ or *NE.

-3 (*NOTRUN) - end code is any value from 90-99. If specified, relation must be *EQ or *NE.

-4 (*ANY) - end code is any value from 0-99. If specified, relation must be *EQ.

Mode CHAR(10)

Possible values are:

ALLNODES** - all nodes of the conditioning activity must meet the completion criteria before this condition is considered met.SAMENODE** - nodes of the conditioned activity need only wait for the same node of the conditioning activity to meet the completion criteria before this condition is considered met.**Number of conditions**

INPUT BINARY(4)

Number of conditions elements (0-5) specified.

Start time

INPUT; CHAR(40)

The start after and start before times. The structure of the template is:

Start after time

CHAR(10); The time after which this activity may be started. The values are:

***CURRENT**

Time specified in HHMMSS format.

Start after date

CHAR(10); The date after which this activity may be started. The values are:

QNSADDCM API

*CURRENT

*NEXT

Date specified in CYYMMDD format.

Start before time

CHAR(10); The time after which this activity must be started. If the activity cannot be started before this time, then it is never started. The values supported are:

*ANY

*CURRENT

Time specified in HHMMSS format.

Start before date

CHAR(10); The date before which this activity must be started. If the activity cannot be started before this date, then it is never started. The values supported are:

*ANY

*CURRENT

*NEXT

Date specified in CYYMMDD format.

Hold

INPUT; CHAR(10)

Specifies whether or not this activity is placed in held status when it is submitted. The values supported are: *YES or *NO.

Replace

INPUT; CHAR(10)

Specifies whether or not this activity should be replaced if it already exists for the change request description. The values supported are: *YES or *NO. When replacing an activity, none of the previous activity information is retained.

Text

INPUT; CHAR(50)

Specifies a brief description of the activity. A special value of *GEN indicates that a description should be generated based on the action and the class of the data object. If the data object class is not recognized by the OS/400 system, the description is blank. The text descriptions generated are:

- Send object
- Retrieve object
- Delete object
- Send PTF
- Retrieve PTF
- Delete PTF
- Apply PTF
- Send and apply PTF
- Remove PTF
- Send and run procedure
- Restart system
- Install product
- Send product
- Send and install product
- Retrieve product

- Delete product
- Install
- Uninstall
- Send and install object
- Send license key

Error code

I/O; CHAR(*)

The structure in which to return error information.

Optional Parameter Group**Activity added**

OUTPUT; CHAR(10)

The name of the activity added to the change request description.

This field is useful when *GEN is specified on the Activity parameter. It provides a method of returning the generated name.

Usage Notes

- To apply PTFs, the verbs install (*INS) or send and install (*SNDINS) should be used.
- The Uninstall (*UNINS) verb can only be performed on installable objects in the iSeries server and on NetView Distribution Manager (NVD) change files (when the target system is running NetView/DM2, Netview DM/6000 or NetView DM/MVS).

Error Messages

If the format of the error code structure provided is not correct, a CPF3CF1 exception message is sent.

CPF24B4 E

Severe error while addressing parameter list.

CPF3CF1 E

Error code parameter not valid.

CPF96A4 E

Activity not added, limit exceeded.

CPF968A E

Activity name &1 not valid.

CPF968C E

Replace value &1 not valid.

CPF968E E

Condition list or start time cannot be specified.

CPF9681 E

Activity &1 already exists.

CPF9682 E

Element &3 of destination array not valid.

CPF9683 E

Number of conditions &1 not valid.

QNSADDCM API

- CPF9684 E
Start after time &1 not valid.
- CPF9685 E
Start before time &1 not valid.
- CPF9686 E
Destination format value &1 not valid.
- CPF9687 E
Number of destinations &1 not valid.
- CPF9688 E
Element &3 of condition list array not valid.
- CPF9689 E
Hold value &1 not valid.
- CPF969F E
Activity &1 already exists in condition list.
- CPF9691 E
Start after date &1 not valid.
- CPF9692 E
Start before date &1 not valid.
- CPF9693 E
Activity &1 not found.
- CPF9696 E
Generated activity name limit exceeded.
- CPF9697 E
Activity cannot be conditioned on itself.
- CPF9698 E
Maximum size of CRQ description &1 exceeded.
- CPF9699 E
Start time not valid.
- CPF9801 E
Object &2 in library &1 not found.
- CPF9802 E
Not authorized to object &2 in &3.
- CPF9803 E
Cannot allocate object &2 in library &3.
- CPF9804 E
Object &2 in library &3 damaged.
- CPF9807 E
One or more libraries in library list deleted.
- CPF9808 E
Cannot allocate one or more libraries on library list.
- CPF9810 E
Library &1 not found.
- CPF9820 E
Not authorized to use library &1.

CPF9830 E
Cannot assign library &1.

CPF9838 E
User profile storage limit exceeded.

CPF9872 E
Program or service program &1 in library &2 ended. Reason code &3.

MSS01DA E
Version attribute &4 for token &3 not valid.

MSS01DB E
Generable attribute &4 for token &3 not valid.

MSS01DE E
Match flag &2 not valid.

MSS01DF E
Match attribute &2 not valid.

MSS01D4 E
Match flag &3 not valid.

MSS01D5 E
Match attribute &4 for token &3 not valid.

MSS01D6 E
Length of global name token &3 not valid.

MSS01D7 E
Value of global name token &3 not valid.

MSS01D8 E
Global name not valid.

MSS01EA E
Target release in global name not supported

MSS01EF E
Token attribute &1 cannot be blank.

MSS0116 E
Maximum global name length exceeded.

MSS0117 E
Global name token &3 not valid. Reason code &4.

MSS0119 E
Global name not valid.

MSS014B E
Version attribute &2 not valid.

MSS014C E
Generable attribute &2 not valid.

MSS0142 E
Number of tokens &1 not valid.

MSS0143 E
Match flag &3 not valid.

SMU1691 E
Action &1 not valid.

QNSADDCM API

- SMU1694 E
Action value list length &1 not valid.
- SMU180A E
Parameter ID *CMPALG not valid. Reason code &1.
- SMU180B E
Parameter ID &1 not valid.
- SMU180C E
Offset &1 is outside action value list.
- SMU180D E
Parameter ID required for action &4.
- SMU180E E
Value &1 not valid for data object class.
- SMU180F E
Object type &1 not supported.
- SMU1801 E
Parameter ID &1 not valid.
- SMU1802 E
Parameter ID &1 not valid for Action &2.
- SMU1803 E
Parameter ID &1 specified more than once.
- SMU1804 E
Parameter ID &1 or &2 required.
- SMU1805 E
Value &2 not valid. Reason code &3.
- SMU1806 E
Parameter identifiers cannot be specified together.
- SMU1807 E
Value &3 for parameter ID &1 not valid.
- SMU1808 E
Value &4 for parameter ID &1 not valid.
- SMU1809 E
Date for parameter ID &1 not valid.
- SMU1811 E
Target Release &1 not valid.
- SMU1813 E
Data object class not valid. Reason code &3.
- SMU182A E
Parameter ID &1 required.
- SMU182B E
Parameter ID *CMPALG required.
- SMU182D E
Compression not supported.
- SMU182E E
Member *ALL not valid with object type *FILEDATA.

- SMU182F E**
Member *ALL cannot be specified for object type *FILE.
- SMU1824 E**
Parameter ID required for action &5.
- SMU183A E**
Parameter ID *PTFPART cannot be specified.
- SMU183D E**
Library QTEMP not valid for parameter ID *OBJID.
- SMU183E E**
Action not valid for object type &1.
- SMU183F E**
Parameter ID *ACTION required.
- SMU184A E**
Action *RTV not valid.
- SMU184D E**
Destruction *YES or *ALLOWED required.
- SMU184E E**
Combination of values for parameter IDs not valid.
- SMU184F E**
Release &1 not in valid format.
- SMU1849 E**
Global name not valid for *RTV action
- SMU1851 E**
Cover letter language required.
- SMU1852 E**
Product identifier *ONLY not valid.
- SMU1853 E**
PTF Identifier *ALL not valid for action &1.
- SMU1854 E**
PTF part *CVRLTR not valid for action &1.
- SMU1855 E**
PTF part *ALL not valid for action &1.
- SMU186B E**
Cover letter language cannot be specified.
- SMU186C E**
Product &1 not valid for action &2.
- SMU186D E**
Destruction *NO not valid.
- SMU1871 E**
Global name and object category do not match
- SMU1872 E**
Parameter ID &3 not valid for object category.
- SMU1873 E**
Action &1 not valid for object category.

QNSADDCM API

- SMU1874 E
Object category and ID *PTFFPART do not match.
- SMU1876 E
Member *FIRST or *LAST not valid for *INI.
- SMU188B E
Product option &1 not valid.
- SMU188C E
Load ID &1 not valid.
- SMU188E E
Load ID &1 not valid for load type &2.
- SMU188F E
Cover letter language &1 not valid.
- SMU1895 E
CRQD library &1 not valid.
- SMU1898 E
Removability *NO or *DESIRED required.
- SMU1899 E
Special value for text parameter not valid.
- SMU18B1 E
Number of parameters in action value list not valid.
- SMU18B2 E
Member type only allowed when object type is *FILEDATA.
- SMU18B3 E
Object name value is more than 8 characters.
- SMU18B4 E
Product identifiers do not match.
- SMU18B5 E
Parameter IDs &1 and &2 required.
- SMU18B6 E
License term &1 not valid.
- SMU18B7 E
Component name not valid. Reason code &2.
- SMU18B8 E
Parameter identifiers cannot be specified together.
- SMU18BA E
Component name not valid.
- SMU18BC E
Component name token &3 not valid. Reason code &2.
- SMU18BD E
Product ID not valid.
- SMU18BE E
Parameter ID required for action &6.

Send Change Request Message (QNSSNDCM) API

Parameters

Required Parameter Group:

1	Change request	Input	CHAR(10)
2	Sequence number	Input	BINARY (4)
3	Activity	Input	CHAR(10)
4	Node	Input	CHAR(20)
5	Message file	Input	CHAR(20)
6	Message ID	Input	CHAR(7)
7	Message data	Input	CHAR(*)
8	Message data length	Input	BINARY(4)
9	Error code	I/O	CHAR (*)

The Send Change Request Message (QNSSNDCM) API is used to log messages pertaining to a submitted change request. This provides a way for the application to log intermediate status, error, or completion information (messages) in the change request. The messages can be viewed through the Display Change Request Messages display.

Authorities and Locks

You must either be the user under which the change request is running or have *JOBCTL special authority.

You must have *USE authority to the message file.

Note: If the message to log applies to all *nodes*, but only one message is desired, the node parameter must be *ALL. If the message applies to all *activities*, but only one message is desired, the activity and node parameter must be *ALL.

Required Parameter Group

Change request

INPUT; CHAR(10)

The name of the change request in which the specified activity is contained.

Sequence number

INPUT; BINARY(4)

The sequence number uniquely identifies a change request name. Valid values range from 1 through 999999.

Activity

INPUT; CHAR(10)

The name of the activity for which a message is logged. Special value: *ALL. If *ALL is specified, the message is associated with all activities of the change request.

Node

INPUT; CHAR(20)

QNSSNDCM API

The managed system of the activity for which the message is to be added. A managed system is identified by an APPN network identifier and a control point name.

Special values are:

***LCL** When *LCL is specified, the message is associated with the activity for the local system only.

***ALL** When *ALL is specified, the message is associated with all nodes of the activity.

The structure is:

Network identifier

CHAR(8)

Control point name

CHAR(8)

Reserved

CHAR(4)

Message file

INPUT; CHAR(20)

The name of the message file containing the message ID. Format is:

Name CHAR(10)

Library

CHAR(10). Possible values: *LIBL, *CURLIB, and library name.

Message ID

INPUT; CHAR(7)

The message ID of the message to be logged for the change request.

Message data

INPUT; CHAR(*)

The replacement text associated with the message ID to be logged.

Message data length

INPUT; BINARY(4)

The total length of the replacement text. The maximum size is 512 bytes.

Error code

I/O; CHAR(*)

The error code. The structure in which to return error information.

Error Messages

The following messages are possible as either escape messages or return codes.

CPF24B4 E

Severe error while addressing parameter list.

CPF2401 E

Not authorized to library &1.

CPF2407 E

Message file &1 in &2 not found.

- CPF2411 E**
Not authorized to message file &1 in &2.
- CPF2419 E**
Message identifier &1 not found in message file &2 in &3.
- CPF2499 E**
Message identifier &1 not allowed.
- CPF2531 E**
Message file &1 in &2 damaged for &3.
- CPF2548 E**
Damage to message file &1 in &2.
- CPF3CF1 E**
Error code parameter not valid.
- CPF9830 E**
Cannot assign library &1.
- CPF9845 E**
Error occurred while opening file &1.
- CPF9846 E**
Error while processing file &1 in library &2.
- CPF9872 E**
Program or service program &1 in library &2 ended. Reason code &3.
- SMU1612 E**
Not authorized to change request &1 &2.
- SMU1613 E**
Change request &1 &2 not found.
- SMU1641 E**
Value &1 for message data length not valid.
- SMU1642 E**
Activity &3 not found.
- SMU1643 E**
Node &5 &4 not found.
- SMU1644 E**
Value &1 for sequence number not valid.
- SMU165B E**
Value &1 for message library not valid.
- SMU1662 E**
System Manager internal file damaged or not found.

Update Status (QNSUPDST) API

Parameters

Required Parameter Group:

1	Change request	Input	CHAR(10)
2	Sequence number	Input	BINARY(4)
3	Activity	Input	CHAR(10)
4	Node	Input	CHAR(20)
5	Status	Input	CHAR(10)
6	End code	Input	BINARY(4)
7	Error code	I/O	CHAR(*)

The Update Status (QNSUPDST) API is used to update the status of the specified node for an activity to one of the allowed status values. The end code is required only when the value *ENDED is specified for the status parameter.

Authorities and Locks

You must either be the user under which the change request is running or have *JOBCTL special authority.

This API must be called once for each node for which a status change is desired. (*ALL is not supported for the node parameter.)

Required Parameter Group

Change request

INPUT; CHAR(10)

The name of the change request for which to update the status.

Sequence number

INPUT; BINARY(4)

The sequence number uniquely identifies a change request name. The valid values range from 1 through 999999.

Activity

INPUT; CHAR(10)

The name of the activity to be updated.

Node

INPUT; CHAR(20)

The managed system of the activity for which the status will be updated. A special supported value is *LCL. If *LCL is specified, the activity status for the local system is updated. The structure is:

Network ID

CHAR(8)

Control point

CHAR(8)

Reserved

CHAR(4)

Status

INPUT; CHAR(10)

Specifies the new status value of the activity. Possible values are:

***SENT**

The status is changed to Sent.

***NOTSENT**

The status is changed to Notsent.

***RUNNING**

The status is changed to Running.

***ENDING**

The status is changed to Ending.

***ENDED**

The status is changed to Ended.

End code

INPUT; BINARY(4)

Specifies the end code of the activity. This indicates success or failure of the activity and is determined by the application called to run the activity. This parameter is used only when the value *ENDED is specified for the status parameter. Success is defined as an end code in the range 0 through 9. Failure is defined as an end code in the range 10 through 49. The end codes and their messages or definitions are as follows:

- 00** Activity completed successfully.
- 01-09** Activity completed with warning messages.
- 10-29** Activity started running but failure occurred (includes unexpected responses received from exit program).
- 30-39** Activity started running but was ended (activity or job).
 - 30 = ended through *CNTRLD
 - 35 = ended through *IMMED
 - 39 = ended through *FRCFAIL
- 40-49** Error detected by application before performing the function.
- 90-99** Activity was not run because conditions or schedules were not met.
 - 95 = Scheduled start time expired
 - 99 = Conditions cannot be met

Error code

I/O; CHAR(*)

The error code. The structure in which to return error information.

Error Messages

The following messages are possible as either escape messages or return codes.

CPF24B4 E

Severe error while addressing parameter list.

CPF3CF1 E

Error code parameter not valid.

CPF9845 E

Error occurred while opening file &1.

QNSUPDST API

- CPF9846 E**
Error while processing file &1 in library &2.
- SMU160D E**
Journaling for internal files not active.
- SMU1612 E**
Not authorized to change request &1 &2.
- SMU1613 E**
Change request &1 &2 not found.
- SMU1642 E**
Activity &3 not found.
- SMU1643 E**
Node &5 &4 not found.
- SMU1644 E**
Value &1 for sequence number not valid.
- SMU1655 E**
Value &1 for status not valid.
- SMU1656 E**
Status change to &7 not allowed.
- SMU165A E**
Value &1 for end code not valid.
- SMU1662 E**
System Manager internal file damaged or not found.
- SMU166A E**
Internal System Manager object not found or damaged.
- SMU166D E**
Commit operation failed.
- SMU166E E**
Storage limit exceeded.
- SMU166F E**
Internal processing error occurred.

Add Activity Exit Program

Parameters

Required Parameter Group:

1	Change request description	Input	CHAR(20)
2	Activity	Input	CHAR(10)

The Add Activity Exit Program allows an application to add an activity to a change request description for a previously registered activity type. The exit is called when option 1(Add) is selected from the Work with Change Request Description Activities display and the activity type entered is the same as that registered for this exit program.

Add Activity Exit Program

This exit program prompts the user to enter detailed information pertaining to the activity type to be added to the change request description. Then it calls the Add Activity (QFVADDA) API to add the activity to the change request description.

Exception messages received from the exit program are left in the your job log.

An exit program and the activity type it adds to a change request description are defined using the Add Exit Program (QUSADDEP) API.

Required Parameter Group

Change request description

INPUT; CHAR(20)

The change request description to add an activity to. The format is:

Name CHAR(10)

Library

CHAR(10)

Activity

INPUT; CHAR(10)

The name of the activity to add. This may be blanks.

Notes

The Add Exit Program (QUSADDEP) API mentioned above may be used to register a command rather than a program. If this is the case, the following parameters are required and the values specified must be supported.

CRQD

Change request description object name.

The possible library values are:

***LIBL:** All of the libraries in the user and system portions of the job's library list are searched.

***CURLIB:** The current library for the job is used to locate the object.

library-name: Only the library named in this parameter is searched.

change-request-description: The name of the change request description object.

ACTIVITY

The identifier of the activity to add to the change request description.

***GEN:** An activity identifier will be generated. The activity ID will be of the form QACTxxxxxx where xxxxxx is a number that is incremented for each activity added.

***LAST:** The activity will be the last to run in the change request.

activity-name: A 10 character activity name.

The command is called in the same way as the exit program described above. The difference is that the command prompt display is shown allowing the user to specify detailed information pertaining to the activity type to be added to the change request description. The command calls the Add Activity (QFVADDA) API to add the activity to the change request description.

Add Activity Exit Program

Error Messages

All messages signaled by the exit program are signaled again by calling System Manager program.

Change Activity Exit Program

Parameters

Required Parameter Group:

1	Change request description	INPUT	CHAR(20)
2	Activity	INPUT	CHAR(10)

The Change Activity Exit Program allows an application to change an activity in a change request description. The exit is called when option 2 (Change) is selected from the Work with Change Request Description Activities display and the activity type is the same as that registered for this exit program.

This exit program calls the Retrieve Change Request Description (QFVRTVCD) API to retrieve the current information for the activity. Then the exit program displays the current information and prompts the user to make the desired changes. Then the exit calls the Add Activity (QFVADDA) API with the replace option specified to replace the existing activity in the change request description.

Exception messages received from the exit program are left in the job log.

An exit program and the activity type it changes in a change request description are defined using the Add Exit Program (QUSADDEP) API.

Required Parameter Group

Change request description

INPUT; CHAR(20)

Changes an activity in the change request description specified. The format is:

Name CHAR(10)

Library

CHAR(10)

Activity

INPUT; CHAR(10)

The name of the activity to change.

Error Messages

All messages signaled by the exit program are signaled again by calling System Manager program.

Display Activity Exit Program

Parameters			
Required Parameter Group:			
1	Change request description	Input	CHAR(20)
2	Activity	Input	CHAR(10)
3	Activity type	Input	CHAR(10)
4	Change request	Input	CHAR(10)
5	Sequence number	Input	BINARY(4)
6	Action	Input	CHAR(10)
7	Printer device file	Input	CHAR(20)
8	Node	Input	CHAR(20)
9	Function parameters	Input	CHAR(*)
10	Function parameters length	Input	BINARY(4)
11	Text	Input	CHAR(50)

This exit program allows an application to display or print type-specific information pertaining to an activity in a change request description or submitted change request. The exit is called when you select one of the following:

1. F11=Display type specific data from the Display Change Request Description Activity Details display and the activity type is the same as that registered for this exit program.
2. F11=Display type specific data from the Display Submitted Change Request Activity Details display and the activity type is the same as that registered for this exit program.
3. Option 6=Print from the Work with Change Request Descriptions display and there are activities in the change request description whose activity type is the same as that registered for this exit program.

When a display action is requested, this exit program displays the type-specific information associated with the activity. This information is contained within the function parameters.

When a print action is requested, this exit program calls the Open Print Application (QUIOPNPA) API with the share-open-data path option specified. Then the exit program uses the Print Panel (QUIPRTP) API to print the type-specific information associated with the activity. This information is contained within the function parameters.

Exception messages received from the exit program are left in the job log.

An exit program and the activity type it displays or prints are defined using the Add Exit Program (QUSADDEP) API.

Required Parameter Group

Change request description
INPUT; CHAR(20)

Display Activity Exit Program

The change request description containing the activity to display. If the display is for a change request activity, this parameter is set to *NONE. Otherwise the format is:

Name CHAR(10)

Library
CHAR(10)

Activity

INPUT; CHAR(10)

The name of the activity to display.

Activity type

INPUT; CHAR(10)

Indicates the type of activity to display or print for the change request description. This is passed to the exit program to allow it to validate that it is the correct program for the activity type selected. The following activity types are already registered and should not be registered again.

Name Description

***CHGMGT**
Change management generic activity

***OBJ** Object distribution

***PTF** PTF distribution

***RSC** Resource activation

***PRD** Product distribution

***CMD** Run a command

***LIC** License distribution

Change request

INPUT; CHAR(10)

The name of the change request containing the activity to display. If the display is for a change request description activity, this parameter is set to *NONE.

Sequence number

INPUT; BINARY(4)

The sequence number that uniquely identifies a change request name. If the display is for a change request description activity, this parameter is set to 0 (zero).

Action

INPUT; CHAR(10)

Indicates whether this is a display or print request. The possible values are:

*DISPLAY

*PRINT

Printer device file

INPUT; CHAR(20)

The qualified name of the printer device file to print the application-specific information in. If *DISPLAY is specified for the action parameter, the printer device file parameter is ignored. The format is:

Display Activity Exit Program

Name CHAR(10)

Library
CHAR(10)

Node
INPUT; CHAR(20)

The specific node of the activity to display information about. The structure is:

Network ID
CHAR(8)

Control point
CHAR(8)

Reserved
CHAR(4)

Note: Special values: *ALL (the information to display pertains to all nodes of the activity), *LCL (the information to display pertains to the local node).

Function parameters
INPUT; CHAR(*)

This is a structure containing the activity type-specific information. The format is known only to the registered exit program.

Function parameters length
INPUT; BINARY(4)

The total length of the function parameters parameter. The maximum value is 12288.

Text
INPUT; CHAR(50)

Specifies a brief description of the activity (in the CCSID of the current job).

Error Messages

All messages signaled by the exit program will be signaled again by calling the System Manager program.

Run Activity Exit Program

Parameters

Required Parameter Group:

1	Change request	Input	CHAR(10)
2	Sequence number	Input	BINARY(4)
3	Activity	Input	CHAR(10)
4	Activity type	Input	CHAR(10)
5	Nodes	Input	ARRAY OF CHAR(20)
6	Number of nodes	Input	BINARY(4)
7	Action	Input	CHAR(10)
8	Function parameters	Input	CHAR(*)
9	Function parameters length	Input	BINARY(4)

The Run Activity Exit Program is called to do one of the following:

- Perform™ the function specified by an activity in a change request description.
- Restart the function specified by an activity at an appropriate point. A restart may be required due to a previous interruption.
- End the function specified by an activity. The end request can be for a controlled end, an immediate end, or a force failure end.

The exit program is called when an activity is ready to run or when the End Submitted Change Request Activity (ENDSBMCRQA) command is called.

This exit program is called with the *END action only if the activity has a status of *Active*.

Exception messages received from the exit program are left in your job log. If an exception message is received, the status of the nodes for the activity is changed to 'Not started'.

An exit program and the activity type it runs are defined using the Add Exit Program (QUSADDEP) API.

Required Parameter Group

Change request

INPUT; CHAR(10)

The name of the change request for which an activity is run.

Sequence number

INPUT; BINARY(4)

The sequence number uniquely identifies a change request name.

Activity

INPUT; CHAR(10)

The name of the activity to run.

Activity type

INPUT; CHAR(10)

Indicates the type of activity to run. This is passed to the exit program to allow it to validate that it is the correct program for the activity type selected. The following activity types are already registered and should not be registered again.

Name Description

- *CHGMGT**
Change management generic activity
- *OBJ** Object distribution
- *PTF** PTF distribution
- *RSC** Resource activation
- *PRD** Product distribution
- *CMD** Run a command
- *LIC** License distribution

Nodes

INPUT; ARRAY OF CHAR(20)

The list of node names the activity is targeted for. This is an array with *number of nodes* elements. The structure is:

Network identifier

CHAR(8)

Control point name

CHAR(8)

Reserved

CHAR(4)

Number of nodes

INPUT; BINARY(4)

The number of nodes this activity is targeted for. The value must be greater than or equal to 1.

Action

INPUT; CHAR(10)

Specifies the action mode in which the application is called. The possible values are:

*START

This value is passed the first time the exit program is called.

*RESTART

This value is passed in a recovery situation. The System Manager licensed program was ended somehow and then restarted by the Start System Manager (STRSYSMGR) command. System Manager calls the run activity exit program again for all activities that were in progress. The application must determine if this is a duplicate request or not, and either process it or throw it away.

Run Activity Exit Program

*ENDCNTRLD

This value is passed to indicate that the application should attempt to end the specified activity or activities in a controlled manner. The end code should be set to 30.

*ENDIMMED

This value is passed to indicate that the application should end the specified activities immediately. The end code should be set to 35.

*ENDFRC

This value is passed to indicate that the application should end the specified activities immediately. The status of the activities is forced to Ended and the end code is set to 39.

Function parameters

INPUT; CHAR(*)

This is a structure containing the activity type-specific information. The format is known only to the registered exit program.

Function parameters length

INPUT; BINARY(4)

The total length of the function parameters parameter. The maximum value is 12288 bytes.

Error Messages

SMU1841 E

&4 nodes ended; &5 not ended; &6 being ended.

Action Value List

The action value list will have the following form:

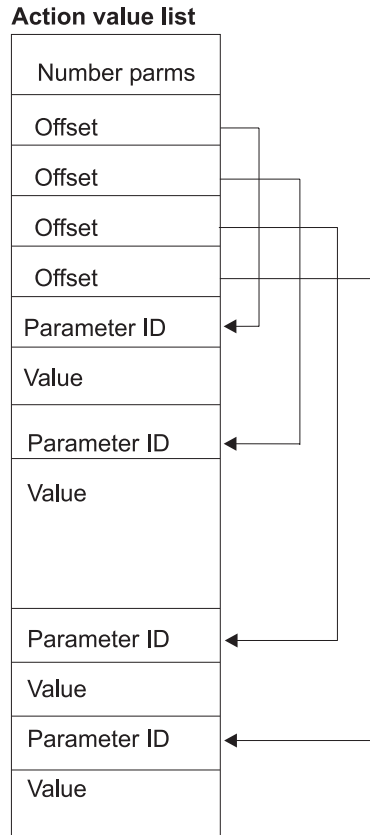
Name	Type	Description
Number of parameters	BIN(4)	Number of offsets (which is the same as number of parameters in pairs in the list) value: from 0 to 50
Offset to Parameter ID	BIN(4)	Offset to first Parameter ID
Offset to Parameter ID	BIN(4)	Offset to next Parameter ID

Note: Offsets are calculated from the beginning of the action value list structure.

The following parameter information is repeated for each parameter.

Parameter ID	CHAR(12)	Identifies the parameter
Value	CHAR(*)	Indicates the value or data for this parameter

Figure 39 on page 317 shows the values that may be specified in the action value list.



rbaep502-0

Figure 39. Example of Action Value List

Notes:

- Parameters identified by (1) are mutually exclusive for the action specified in the action parameter. Only one of these parameters can be specified for each action.
- The parameter ID, action, and structure are always required on the action value list.
- In the action value list, if a PTF_IDENTIFICATION, an OBJECT_IDENTIFICATION, or a PRODUCT_IDENTIFICATION value is used, a global name cannot be specified.

Here is a list of valid parameters for each action:

Action	Accept (*ACC)		
	Use	Reference	
		See	Page
action	required	ACTION	323
data object class	optional	DATA_OBJECT_CLASS	329
execute after time	optional	EXECUTE_AFTER_TIME	331
stored name (1)	required	STORED_NAME	338
PTF identification (1)	required	PTF_IDENTIFICATION	336
PTF part	optional	PTF_Part	336

Run Activity Exit Program

user	optional	USER	340
------	----------	------	-----

Action	Activate (*ACT)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
activation force	optional	ACTIVATION_FORCE	324
activation use on activate	optional	ACTIVATION_USE_ON_ACTIVATE	324
configuration name	optional	CONFIGURATION_NAME	328
execute after time	optional	EXECUTE_AFTER_TIME	331
user	optional	USER	340

Action	Delete (*DLT)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
data object class	optional	DATA_OBJECT_CLASS	329
deleting match flags	optional	DELETING_MATCH_FLAGS	330
to be deleted name (1)	required	TO_BE_DELETED_NAME	338
object identification (1)	required	OBJECT_IDENTIFICATION	333
PTF identification (1)	required	PTF_IDENTIFICATION	336
PTF part	optional	PTF_Part	336
product identification (1)	required	PRODUCT_IDENTIFICATION	335
user	optional	USER	340

Action	Initiate (*INI)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
data object class	optional	DATA_OBJECT_CLASS	329
execute after time	optional	EXECUTE_AFTER_TIME	331
initiate parameters	optional	INITIATE_PARAMETERS	332
object disposition	optional	OBJECT_DISPOSITION	333
stored name (1)	required	STORED_NAME	338
object identification (1)	required	OBJECT_IDENTIFICATION	333
user	optional	USER	340

Run Activity Exit Program

Action	Install (*INS)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
activation use on install	optional	ACTIVATION_USE_ON_INSTALL	325
alter active components	optional	ALTER_ACTIVE_COMPONENTS	325
auto accept	optional	AUTO_ACCEPT	325
auto remove	optional	AUTO_REMOVE	326
corequisite change name list	optional	COREQUISITE_CHANGE_NAME_LIST	329
data object class	optional	DATA_OBJECT_CLASS	329
execute after time	optional	EXECUTE_AFTER_TIME	331
object disposition	optional	OBJECT_DISPOSITION	333
posttest	optional	POSTTEST	334
pretest	optional	PRETEST	334
removability	optional	REMOVABILITY	337
stored name (1)	required	STORED_NAME	338
PTF identification (1)	required	PTF_IDENTIFICATION	336
PTF part	optional	PTF_Part	336
product identification (1)	required	PRODUCT_IDENTIFICATION	335
user	optional	USER	340

Action	Remove (*RMV)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
alter active components	optional	ALTER_ACTIVE_COMPONENTS	325
data object class	optional	DATA_OBJECT_CLASS	329
execute after time	optional	EXECUTE_AFTER_TIME	331
posttest	optional	POSTTEST	334
stored name (1)	required	STORED_NAME	338
PTF identification (1)	required	PTF_IDENTIFICATION	336
PTF part	optional	PTF_Part	336
user	optional	USER	340

Action	Retrieve (*RTV)		
Parameter	Use	Reference	
		See	Page

Run Activity Exit Program

action	required	ACTION	323
compression algorithm	optional	COMPRESSION_ALGORITHM	327
compression store state	optional	COMPRESSION_STORE_STATE	328
compression transfer state	optional	COMPRESSION_TRANSFER_STATE	328
data object class	optional	DATA_OBJECT_CLASS	329
destruction	optional	DESTRUCTION	330
fetching match flags	optional	FETCHING_MATCH_FLAGS	331
to be fetched name (1)	required	TO_BE_FETCHED_NAME	338
object identification (1)	required	OBJECT_IDENTIFICATION	333
PTF identification (1)	required	PTF_IDENTIFICATION	336
PTF part	optional	PTF_Part	336
product identification (1)	required	PRODUCT_IDENTIFICATION	335
user	optional	USER	340

Action	Send (*SND)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
compression algorithm	optional	COMPRESSION_ALGORITHM	327
compression store state	optional	COMPRESSION_STORE_STATE	328
compression transfer state	optional	COMPRESSION_TRANSFER_STATE	328
data object class	optional	DATA_OBJECT_CLASS	329
deleting match flags	optional	DELETING_MATCH_FLAGS	330
destruction	optional	DESTRUCTION	330
fetching match flags	optional	FETCHING_MATCH_FLAGS	331
to be deleted name	optional	TO_BE_DELETED_NAME	338
to be fetched name (1)	required	TO_BE_FETCHED_NAME	338
token attributes list	optional	TOKEN_ATTRIBUTES_LIST	339
object identification (1)	required	OBJECT_IDENTIFICATION	333
PTF identification (1)	required	PTF_IDENTIFICATION	336

Run Activity Exit Program

PTF part	optional	PTF_Part	336
product identification (1)	required	PRODUCT_IDENTIFICATION	335
user	optional	USER	340

Action	Send and Initiate (*SNDINI)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
compression algorithm	optional	COMPRESSION_ALGORITHM	327
compression store state	optional	COMPRESSION_STORE_STATE	328
compression transfer state	optional	COMPRESSION_TRANSFER_STATE	328
data object class	optional	DATA_OBJECT_CLASS	329
deleting match flags	optional	DELETING_MATCH_FLAGS	330
destruction	optional	DESTRUCTION	330
execute after time	optional	EXECUTE_AFTER_TIME	331
fetching match flags	optional	FETCHING_MATCH_FLAGS	331
initiate parameters	optional	INITIATE_PARAMETERS	332
object disposition	optional	OBJECT_DISPOSITION	333
to be deleted name	optional	TO_BE_DELETED_NAME	338
to be fetched name (1)	required	TO_BE_FETCHED_NAME	338
token attributes list	optional	TOKEN_ATTRIBUTES_LIST	339
object identification (1)	required	OBJECT_IDENTIFICATION	333
user	optional	USER	340

Action	Uninstall (*UNINS)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
alter active components	optional	ALTER_ACTIVE_COMPONENTS	325
component change name	required	COMPONENT_CHANGE_NAME	329
execute after time	optional	EXECUTE_AFTER_TIME	331
force indication	optional	FORCE_INDICATION	326
refresh change level	optional	REFRESH_CHANGE_LEVEL	337

Run Activity Exit Program

user	optional	USER	340
------	----------	------	-----

Action	Send and Install (*SNDINS)		
Parameter	Use	Reference	
		See	Page
action	required	ACTION	323
activation use on install	optional	ACTIVATION_USE_ON_INSTALL	325
alter active components	optional	ALTER_ACTIVE_COMPONENTS	325
auto accept	optional	AUTO_ACCEPT	325
auto remove	optional	AUTO_REMOVE	326
compression algorithm	optional	COMPRESSION_ALGORITHM	327
compression store state	optional	COMPRESSION_STORE_STATE	328
compression transfer state	optional	COMPRESSION_TRANSFER_STATE	328
corequisite change name list	optional	COREQUISITE_CHANGE_NAME_LIST	329
data object class	optional	DATA_OBJECT_CLASS	329
deleting match flags	optional	DELETING_MATCH_FLAGS	330
destruction	optional	DESTRUCTION	330
execute after time	optional	EXECUTE_AFTER_TIME	331
fetching match flags	optional	FETCHING_MATCH_FLAGS	331
object disposition	optional	OBJECT_DISPOSITION	333
posttest	optional	POSTTEST	334
pretest	optional	PRETEST	334
removability	optional	REMOVABILITY	337
to be deleted name	optional	TO_BE_DELETED_NAME	338
to be fetched name (1)	required	TO_BE_FETCHED_NAME	338
token attributes list	optional	TOKEN_ATTRIBUTES_LIST	339
PTF identification (1)	required	PTF_IDENTIFICATION	336
PTF part	optional	PTF_Part	336
product identification (1)	required	PRODUCT_IDENTIFICATION	335
user	optional	USER	340

Parameter Descriptions

This section contains the description of the parameters value list.

Note: Offsets within a container are calculated from the beginning of that container (for example, from the containers parameter ID).

ACTION

Specifies the function to be performed by the managed system and the object category to run the action against.

Name	Type	Length	Valid Values / Description																								
Parameter ID	Char	12	*ACTION																								
Value																											
Action	Char	10	<table border="0"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>*ACC</td> <td>Accept</td> </tr> <tr> <td>*ACT</td> <td>Activate</td> </tr> <tr> <td>*DLT</td> <td>Delete</td> </tr> <tr> <td>*INI</td> <td>Initiate</td> </tr> <tr> <td>*INS</td> <td>Install</td> </tr> <tr> <td>*RMV</td> <td>Remove</td> </tr> <tr> <td>*RTV</td> <td>Retrieve</td> </tr> <tr> <td>*SND</td> <td>Send</td> </tr> <tr> <td>*SNDINI</td> <td>Send and initiate</td> </tr> <tr> <td>*SNDINS</td> <td>Send and install</td> </tr> <tr> <td>*UNINS</td> <td>Uninstall</td> </tr> </tbody> </table>	Value	Description	*ACC	Accept	*ACT	Activate	*DLT	Delete	*INI	Initiate	*INS	Install	*RMV	Remove	*RTV	Retrieve	*SND	Send	*SNDINI	Send and initiate	*SNDINS	Send and install	*UNINS	Uninstall
Value	Description																										
*ACC	Accept																										
*ACT	Activate																										
*DLT	Delete																										
*INI	Initiate																										
*INS	Install																										
*RMV	Remove																										
*RTV	Retrieve																										
*SND	Send																										
*SNDINI	Send and initiate																										
*SNDINS	Send and install																										
*UNINS	Uninstall																										

Default:

- None, required parameter

Dependencies:

- Object category OBJ cannot be specified when the Action is *ACC, *ACT, *INS, *RMV, *SNDINS, or *UNINS.
- Object category PTF cannot be specified when the Action is *ACT, *INI, *SNDINI, or *UNINS.
- Object category CVRLTR cannot be specified when the Action is *ACC, *ACT, *DLT, *INI, *INS, *RMV, *SNDINI, *SNDINS, or *UNINS.
- Object category PRODUCT cannot be specified when the Action is *ACC, *ACT, *INI, *RMV, *SNDINI, or *UNINS.
- Object category NONE can be specified only when the Action is *ACT.
- Object category NONE can only be specified when the Action is *ACT.
- Object category ANY cannot be specified when the Action is *ACT.
- Object category ANY can only be specified when the parameter ID *STOREDN, *TODLTN, *TOFTCHN or *COMPNAME is also specified.
- Object category LICKEY can only be specified when the Action is *SND

Run Activity Exit Program

You must specify the LICENSE_IDENTIFICATION parameter that includes license term and product information.

- Object category PRDLICKEY can only be specified when the Action is *SND or *SNDINS.

You must specify the PRODUCT_IDENTIFICATION and the LICENSE_IDENTIFICATION parameters that includes license term and product information.

ACTIVATION_FORCE

Specifies whether or not the target system should proceed with the activation based on the current processing or operations.

In either case, the managed system may reply with activation acceptance before activation is attempted.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*ACTFRC
Value			
Activation_Force	Char	10	Constants: • *NO • *YES
Delay units	Char	10	Constants: • *SECONDS • *MINUTES • *HOURS
Delay period	Binary	4	1-65535. Specifies the maximum amount of time that the managed system may wait before taking the action specified.

Default:

- Activation_Force = *NO
- Delay units = *SECONDS
- Delay period = 3600

Dependencies:

- Delay Units and Delay Period are ignored when Activation_Force is *YES.
- If Activation_Force is *NO and Delay Period is zero, then Activation_Force *YES is performed.

ACTIVATION_USE_ON_ACTIVATE

Specifies which components altered by changes are used during the activation.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*ACTUACT
Value			

Run Activity Exit Program

Activation_Use_On_Activate	Char	10	Constants: <ul style="list-style-type: none"> • *BOTH - Both trial and production • *PROD - Production only • *LAST - Last used, either both trial and production or production only
----------------------------	------	----	--

Default:

- None, this is not required by the architecture

ACTIVATION_USE_ON_INSTALL

Specifies whether the component to be altered by the installation process will be trial version or production version.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*ACTUINS
Value			
Activation_Use_On_Install	Char	10	Constants: <ul style="list-style-type: none"> • *TRIAL - Trial only • *PROD - Production only

Default:

- Activation_Use_On_Install = *PROD

Dependencies:

- If Activation_Use_On_Install is *TRIAL, then Removability must be *YES.

ALTER_ACTIVE_COMPONENTS

Specifies whether or not the target system is allowed to apply the component alterations to the active system; if not, then such action is to be deferred until the next activate.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*ALTACTCMP
Value			
Alter_Active_Components	Char	10	Constants: <ul style="list-style-type: none"> • *ALLOWED - Allowed • *NOTALLOWED - Not allowed

Default:

- None, this is not required by the architecture

AUTO_ACCEPT

Specifies whether the target system accepts objects automatically if installation and any test performed are successful, in order to release resources required to maintain removability as soon as possible. Like a separate accept request, the target system deletes the objects after successful automatic acceptance.

Name	Type	Length	Valid Values / Description
------	------	--------	----------------------------

Run Activity Exit Program

Parameter ID	Char	12	*AUTACC
Value			
Auto_Accept	Char	10	Constants: <ul style="list-style-type: none"> • *YES • *DESIRED • *NO
reserved	Char	2	
Delay	Binary	4	0-255, Specifies the number of days the entry point is expected to wait before accepting the object automatically.

Default:

- Auto_Accept = *NO
- Delay = 0

Dependencies:

- If Removability is *NO, then Auto_Accept cannot be specified.
- Auto_Accept cannot be *YES if Removability is *DESIRED.
- Delay is ignored when Auto_Accept is *NO.

AUTO_REMOVE

Specifies whether the target system removes the object automatically if either an installation or a test fails.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*AUTRMV
Value			
Auto_Remove	Char	10	Constants: <ul style="list-style-type: none"> • *YES • *DESIRED • *NO

Default:

- Auto_Remove = *DESIRED

Dependencies:

- If Removability is *NO, then Auto_Remove cannot be specified.
- Auto_Remove cannot be *YES if Removability is *DESIRED

FORCE_INDICATION

Specifies whether to perform the uninstalling even if the software component includes some change file in a pending installation execution or in a not terminal status.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*FORCEIND
Value			

Run Activity Exit Program

Force_Indication	Char	10	Constants: <ul style="list-style-type: none"> • *YES - Allow uninstalling even if there are change files for the software component in pending installation execution. • *NO - Do not allow uninstalling if there are change files for the software component in pending installation execution.
------------------	------	----	--

Default:

- *NO.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*REFLVL
Value			
Refresh_level	Char	Up to 16 digits	It is a numeric character string of up to 16 digits

Default:

- None.

COMPRESSION_ALGORITHM

Specifies the compression algorithm and related information associated with the compression of a particular change object.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*CMPALG
Value			
Compression_Request	Char	10	Constants: <ul style="list-style-type: none"> • *ADAPTIVE • *SNA • *USER
Reserved	Char	2	
Compression_Parameters	Char	*	The next structure depends on which type of compression was requested: <ul style="list-style-type: none"> • *ADAPTIVE - none • *SNA - (See "SNA_COMPRESSION" on page 337) • *USER - (See "USER_COMPRESSION" on page 340)

Default:

- None.

Dependencies:

Run Activity Exit Program

- Parameter ID and structure must be specified when Compression_Store_State is *CMP and Compression_Transfer_State is *DECMP
- Parameter ID and structure cannot be specified when Compression_Store_State is not specified and Compression_Transfer_State is not specified
- Parameter ID and structure cannot be specified when Compression_Store_State is *DECMP and Compression_Transfer_State is not specified
- Parameter ID and structure cannot be specified when Compression_Store_State is not specified and Compression_Transfer_State is *DECMP
- Parameter ID and structure cannot be specified when Compression_Store_State is *DECMP and Compression_Transfer_State is *DECMP

COMPRESSION_STORE_STATE

Specifies whether or not the object should be stored in compressed format at the target system.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*CMPSTRSTT
Value			
Compression_Store_State	Char	10	Constants: <ul style="list-style-type: none">• *DECMP• *CMP

Default:

- None

COMPRESSION_TRANSFER_STATE

Specifies whether or not the object should be transferred to the target system in compressed format.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*CMPTRSTT
Value			
Compression_Transfer_State	Char	10	Constants: <ul style="list-style-type: none">• *DECMP - Decompress• *CMP - Compress

Default:

- None

CONFIGURATION_NAME

Specifies a name identifying a set of resources to which the activation applies.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*CFGNAME
Value			
Configuration_Name	Char	16	Name of a resource. Leading and imbedded space (X'40') characters are not allowed, and trailing space characters are ignored. Configuration_Name must be in 01134-00500 code set.

Default:

- None, this is not required by the architecture

COREQUISITE_CHANGE_NAME_LIST

Specifies a list of global names to be corequisites.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*COREQCHGNL
Value			
COREQUISITE_CHANGE_NAME_LIST	Char	*	See "FS_GLOBAL_NAME_LIST" on page 332

Default:

- None, this is not required by the architecture

Dependencies:

- Minimum number of corequisite change names is 1.
- Maximum number of corequisite change names is 6.

DATA_OBJECT_CLASS

Specifies the classification code of the file.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*DTAOBJCLS
Value			
Class Code 1	Char	1	VALUE ¹
Class Code 2	Char	1	VALUE ¹
Class Code 3	Char	1	VALUE ¹
Class Code 4	Char	1	VALUE ¹
Note: ¹ For the specific values for data object class, see the <i>Managed System Services for AS/400 Use</i> book.			

Default:

- Class Codes 1,2,3,4 = '00' hex

COMPONENT_CHANGE_NAME

Identifies the Software Component name to be involved in the uninstalling process.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*COMPNAME
Value			

Run Activity Exit Program

Component_Change_Name	Char	*	It consists of the SNA file services global name tokens used to specify the change files component name. It is the set of global name tokens before the token with the REF STI. The maximum number of tokens allowed is 7. See FS_GLOBAL_NAME for information on the format of global names.
-----------------------	------	---	--

Default:

- None, this is a required container

DELETING_MATCH_FLAGS

The deleting match flags govern the partial matching operation at delete time.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*DELETEFLAGS
Value			
Number of Deleting Match Flags	Binary	4	1-10.
Deleting Match Flags	Char	10	Match Flags <ul style="list-style-type: none"> • *MATCH - Match specified token value • *IGNORE - Match any value • *HIGHEST - Select highest token value • *LOWEST - Select lowest token value

Default:

- None

DESTRUCTION

Specifies whether or not one or more objects can be destroyed as part of the action requested.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*DESTRUCT
Value			
Destruction	Char	10	Constants: <ul style="list-style-type: none"> *YES Must Replace the object *ALLOWED Replace or Add the object *NO Must Add the object

Default:

- DESTRUCTION = *NO

Dependencies:

- If Destruction is *NO then a To_Be_Deleted_Name cannot be present.

EXECUTE_AFTER_TIME

Specifies the date and time which must pass before the action is to be performed on the managed system, based on its own time-of-day clock.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*EXEAFTERTIME
Value			
Year	Binary	4	1971 - 2070
Month	Binary	4	1-12
Day	Binary	4	1-31
Hours	Binary	4	0-23
Minutes	Binary	4	0-59
Local_or_GMT	Char	10	Constants: <ul style="list-style-type: none"> • *LOCAL - Managed system time (Local system time) • *GMT - Greenwich Mean Time

Default:

- None

FETCHING_MATCH_FLAGS

The fetching match flags govern the partial matching operation at fetch time.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*FETCHFLAGS
Value			
Number of Fetching Match Flags	Binary	4	1-10.
Fetching Match Flags	Char	10	Match Flags <ul style="list-style-type: none"> • *MATCH - Match specified token value • *IGNORE - Match any value • *HIGHEST - Select highest token value • *LOWEST - Select lowest token value

Default:

- None

FS_GLOBAL_NAME

Specifies a global name.

Tokens 2-10 are optional. Tokens must be specified in sequence.

This structure is a subparameter of "FS_GLOBAL_NAME_LIST" on page 332, and is used to define the following parameters:

Run Activity Exit Program

- "STORED_NAME" on page 338
- "TO_BE_DELETED_NAME" on page 338
- "TO_BE_FETCHED_NAME" on page 338

Name	Type	Length	Valid Values / Description
Value			
Number of tokens	Binary	4	1-10
FS_GLOBAL_NAME_TOKEN	Char	16	Global Name Tokens If there are less than 16 characters in the token value, each token field should be right-filled with blanks.

Default:

- None

FS_GLOBAL_NAME_LIST

Specifies a list of global file names.

This structure is used to define the following parameters:

- "COREQUISITE_CHANGE_NAME_LIST" on page 329

Name	Type	Length	Valid Values / Description
Value			
Number of names in list	Binary	4	Number of global names in the list
Offset to FS_GLOBAL_NAME	Binary	4	See "FS_GLOBAL_NAME" on page 331
Offset to FS_GLOBAL_NAME	Binary	4	See "FS_GLOBAL_NAME" on page 331

Default:

- None

LICENSE_ID

INITIATE_PARAMETERS

Specifies the parameters to be passed when initiating an object.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*INITPARMS
Value			
CCSID of Parameter_Data	Binary	4	0-65535. A two byte CCSID as documented in <i>Code Page Architecture - Stage 1, CPA-S1-AD-8909-00</i>
Number of parameters	Binary	4	Value 1 to 40. Number of parameters to be passed

Run Activity Exit Program

Offset to PARAMETER_DATA	Binary	4	Offset to where the PARAMETER_DATA is located in the Value_list. There may be 1 to 40 offsets. See "PARAMETER_DATA" on page 334
-----------------------------	--------	---	--

Default:

- None

OBJECT_DISPOSITION

Specifies whether or not the object should be kept or deleted after the function has been successfully performed.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*OBJDISPOS
Value			
Object_Disposition	Char	10	Constants: <ul style="list-style-type: none"> • *KEEP • *DELETE

Default:

- None, this is not required by the architecture

OBJECT_IDENTIFICATION

Identifies the OS/400 object that is to be distributed.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*OBJID
Value			
Object_Name	Char	10	OS/400 object name
Object_Library	Char	10	Name of the library in which the object is located. Special values (e.g. *LIBL) are not permitted.
Object_Type	Char	10	OS/400 object type
Member_Name	Char	10	Physical file member name if object type is *FILE of *FILEDATA. Constants: <ul style="list-style-type: none"> • *FIRST • *LAST • *ALL (not allowed when *FILEDATA object type)
Member_Type	Char	10	Physical file member type if object type is *FILEDATA. Constants: <ul style="list-style-type: none"> • *UNSPEC • *REXX • *CL

Run Activity Exit Program

Target_Release	Char	10	The object should be saved for a system of this release. Constants: <ul style="list-style-type: none"> • *CURRENT • *PRV • VxRxMx
----------------	------	----	--

Default:

- None, required if global name, PTF_Identification and Product_Identification are not specified.

PARAMETER_DATA

Specifies the parameter data passed when initiating a program.

This structure is a subparameter of "LICENSE_ID" on page 332.

Name	Type	Length	Valid Values / Description
Value			
Length_of_Parameter_Data	Binary	4	From 1 to 253 Length of parameter data
Parameter_Data	Char	*	From 1 to 253 Bytes Hex data passed to program

Default:

- None

POSTTEST

Specifies whether or not the managed system is to perform a test on the objects after installing or removing them.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	* POSTTST
Value			
Posttest	Char	10	Constants: <ul style="list-style-type: none"> • *YES • *DESIRED • *NO

Default:

- Posttest = *DESIRED

PRETEST

Specifies whether or not the target system is to perform a test on the objects before installing them.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	* PRETST
Value			

Run Activity Exit Program

Pretest	Char	10	Constants: <ul style="list-style-type: none"> • *YES • *DESIRED • *NO
---------	------	----	--

Default:

- Pretest = *DESIRED

PRODUCT_IDENTIFICATION

Identifies an OS/400 product that is to be distributed.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*PRODUCTID
Value			
Product_ID	Char	7	Name of an OS/400 product. The product identifier must only contain the characters A-Z or 0-9
Release	Char	10	Release level of product Constants: <ul style="list-style-type: none"> • *ONLY • VxRxMx
Product_Option	Char	10	Optional parts of the Product It can be any number from 0001 through 0099 Constants: <ul style="list-style-type: none"> • *BASE
Load_Type	Char	10	Specifies whether the product load for the product describes a language, code or all loads Constants: <ul style="list-style-type: none"> • *ALL • *CODE • *LNG
Load_ID	Char	10	Specifies the product load for the product that is to be distributed When Load_Type is *CODE the only valid value is *CODE. When Load_Type is *LNG, the valid values must be in the 29XX format Constants: <ul style="list-style-type: none"> • *CODE (It can only be specified when the Load_Type parameter was specified as *CODE) • *ALL (Only valid if Load_Type is either *LNG or *ALL)

Run Activity Exit Program

Target_Release	Char	10	The product was packaged for distribution for a system of this release Constants: <ul style="list-style-type: none"> • *CURRENT • *PRV • VxRxMx • *ONLY
Reserved	Char	3	Reserved for alignment

Default:

- None, required if global name, PTF_Identification and Object_Identification are not specified.

Dependency

When Product_Identification is specified, and the action is *INS or *SNDINS, then Removability must be *NO or *DESIRED.

PTF_IDENTIFICATION

Identifies an OS/400 PTF that is to be distributed.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*PTFID
Value			
PTF_ID	Char	7	OS/400 PTF identifier. Constants: <ul style="list-style-type: none"> • *ALL
Product_Identifier	Char	7	Name of an OS/400 product with which the PTF is associated. Constants: <ul style="list-style-type: none"> • *ONLY
Release	Char	10	Release level of the PTF. Constants: <ul style="list-style-type: none"> • VxRxMx

Default:

- None, required if global name, Object_Identification and Product_Identification are not specified.

PTF_Part

The PTF_Part value is optionally used when a PTF is to be distributed. It can only be specified if PTF_Identification is also specified. It indicates if only the PTF or the PTF cover letter should be sent.

If a service requester entry exists for the system or systems to which the PTF is to be sent, the cover letter language for that system is determined by that entry. If there is no entry the default cover letter language is used.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*PTFPART
Value			

Run Activity Exit Program

PTF_Part	Char	10	Parts of PTF to send. Constants: <ul style="list-style-type: none"> • *PTF • *CVRLTR
Cover_Letter_Language	Char	10	Cover letter language code (29xx). Constants: <ul style="list-style-type: none"> • *SRVRQS

Default:

- PTF_Part = *PTF

REFRESH_CHANGE_LEVEL

REMOVABILITY

Specifies whether or not objects are to be installed in a removable manner (so that a subsequent Remove action can be issued against them).

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	* RMVABLE
Value			
Removability	Char	10	Constants: <ul style="list-style-type: none"> • *YES • *DESIRED • *NO

Default:

- Removability = *YES

Dependencies:

- Removability must be *YES if Activation_Use_On_Install is *TRIAL
- If Removability is *DESIRED, then Auto_Accept cannot be *YES
- If Removability is *DESIRED, then Auto_Remove cannot be *YES
- If Removability is *NO, then Auto_Accept cannot be specified
- If Removability is *NO, then Auto_Remove cannot be specified

SNA_COMPRESSION

Specifies information about the SNA compression algorithm as it pertains to the requested object.

This structure is a subparameter of "COMPRESSION_ALGORITHM" on page 327, and is mutually exclusive with "USER_COMPRESSION" on page 340.

Name	Type	Length	Valid Values / Description
Value			
SNA_Prime_Character	Char	1	The prime compression character to be associated with the single control byte (SCB) used by the SNA compression algorithm. Valid values are '00'x - 'FF'x.

Default:

Run Activity Exit Program

- SNA_Prime_Character = '40'x.

STORED_NAME

Specifies the file name that was stored, or that has already been stored and is now the subject of a SNA/MS report.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*STOREDN
Value			
STORED_NAME	Char	*	See "FS_GLOBAL_NAME" on page 331

Default:

- Zero number of tokens

TO_BE_DELETED_NAME

Specifies the name of the object to be deleted.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*TODLTN
Value			
TO_BE_DELETED_NAME	Char	*	See "FS_GLOBAL_NAME" on page 331

Default:

- Zero number of tokens

Dependencies:

- Parameter ID and structure cannot be present if Destruction is *NO

TO_BE_FETCHED_NAME

Specifies the name of the object to be fetched.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*TOFTCHN
Value			
TO_BE_FETCHED_NAME	Char	*	See "FS_GLOBAL_NAME" on page 331

Default:

- None

TOKEN_ATTRIBUTES

The token attribute defines for each token in the global name how that token can be used in partial matching or token value generation. For more information about tokens, see the *Managed System Services for AS/400 Use* book.

This structure is a subparameter of "TOKEN_ATTRIBUTES_LIST" on page 339.

Name	Type	Length	Valid Values / Description
Value			

Run Activity Exit Program

Match Indicator	Char	10	Constants: <ul style="list-style-type: none"> • *MATCH - Must Match • *NOMATCH - Need Not Match
Generable Indicator	Char	10	Constants: <ul style="list-style-type: none"> • *YES - Generable • *NO - Not Generable
Version Indicator	Char	10	Constants: <ul style="list-style-type: none"> • *UNSPEC - Unspecified • *NETID - Network identifier • *CPNAME - Control Point Name • *SYSTYPE - System Type • *STI - Subtree Indicator • *ORDCHAR - Ordered Character • *ORDDEC - Ordered Decimal • *ORDDATE - Ordered Date • *ORDTIME - Ordered Time
Reserved	Char	2	

Default:

- None

TOKEN_ATTRIBUTES_LIST

The token attribute defines for each token in the global name how that token can be used in partial matching or token value generation.

Tokens 2-10 are optional. Tokens must be specified in sequence.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*TKNATRL
Value			
Number of tokens	Binary	4	1-10.
Token Attribute Array	Char	32	TOKEN_ATTRIBUTES <ul style="list-style-type: none"> • Match Indicator • Generable Indicator • Version Indicator See "TOKEN_ATTRIBUTES" on page 338

Default:

- None

Dependencies:

- Parameter ID and structure are required when using a subtree indicator for a global name with the standard AS/400 object names or when *CVRLTR is specified for the To_Be_Fetched_Name

Run Activity Exit Program

USER

The USER value is optionally used to store information in the change request for a later use. The format of this information is known only by the user.

The user can retrieve the *USER data using the Retrieve Change Request Description (QFVRTVCD) API.

Name	Type	Length	Valid Values / Description
Parameter ID	Char	12	*USER
Value			
User data	Char	512	Data known only by the user

Default:

- None

USER_COMPRESSION

Specifies information about a named user compression algorithm as it pertains to the requested object.

This structure is a subparameter of "COMPRESSION_ALGORITHM" on page 327, and is mutually exclusive with "SNA_COMPRESSION" on page 337.

Name	Type	Length	Valid Values / Description
Value			
User_Algorithm_Name	Char	12	The name of the user compression algorithm that pertains to the requested object.
User_Parameter_CCSID	Binary	4	0-65535. CCSID of User_Parameter.
User_Parameter_Length	Binary	4	1-27. Length of User_Parameter.
User_Parameters	Char	27	User parameters that apply to the user compression algorithm named in the User_Algorithm_Name parameter.

Default:

- None

Dependencies:

- User_Algorithm_Name must be in 00640-00500 code set.
- User_Parameters are not validated against the specified CCSID value.

Other APIs—Used with Products

Retrieving a list of products

You can use the Select Product (QSZSLTPR) API to retrieve a list of installed products or product options. This API can also retrieve other types of lists of products, such as all products that have been supported using the Work with Supported Products (WRKSPTPRD) command.

Displaying a list of products

You can use the Select Product (QSZSLTPR) API to display a list of installed, supported, or other products known to the system. It consists of two views. You determine which view you want displayed first. One view includes the product, product option, and description fields. The other view includes the product, product option, and release fields. You can also define the F3 key to display on the Select Product display.

Retrieving details about a single product

The Retrieve Product Information (QSZRTVPR) API returns information about a software product. The information is requested by specifying a product ID, release level, option number, and load ID; not by specifying an object name.

You can use this API to:

- Retrieve general information about a product load, including whether the product load is installed or not.
- Retrieve the library list of a product load.
- Retrieve the folder list of a product load.
- Retrieve the object list of a product load.
- Retrieve the list of option and load ID pairs that are valid for a product ID and release combination. This is based on what is listed in the product definition (*PRDDFN) for that product ID and release combination.
- Retrieve information from a product definition, including:
 - The copyright information.
 - The release date.
 - The message file name and library.
 - Whether the product allows multiple releases.
 - The message ID for each option.
 - Whether each option allows dynamic naming.
- Retrieve the current release level of the operating system.
- Retrieve the previous release level of the operating system.
- Retrieve a list of valid release levels of the operating system from a given release level through the currently installed release level.

Note: The Retrieve Object Description (QUSROBJD) API can be used to retrieve product information from the object description of an object. The product ID and release level from the object description is returned by QUSROBJD in format OBJD0300.

- Retrieve the primary language ID of a product.

Run Activity Exit Program

Appendix F. Commands shipped with public authority *Exclude

Table 10 identifies which System Manager commands have restricted authorization (public authority *EXCLUDE) when the licensed program is installed. It shows what IBM-supplied user profiles are authorized to use these restricted commands. For more information about IBM-supplied user profiles, see the *iSeries Security Reference* book.

In Table 10, commands that are restricted to the security officer and any user profile with *ALLOBJ authority have an **R** in the QSECOFR profile. Commands that are specifically authorized to one or more IBM-supplied user profiles, in addition to the security officer, have an **S** under the profile names for which they are authorized.

Any System Manager commands not listed here are public, which means they can be used by all users. However, some commands require special authority, such as *SERVICE or *JOBCTL. The special authorities required for a command are listed in the *iSeries Security Reference*.

If you choose to grant other users or the public *USE authority to these commands, update this table to indicate that commands are no longer restricted on your system. Using some commands may require the authority to certain objects on the system as well as to the commands themselves. See the *iSeries Security Reference* book for the object authorities required for commands.

Table 10. Authorities of IBM-Supplied User Profiles to Restricted Commands

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
ADDCMDCRQA		S	S	S	S
ADDCRQA		S	S	S	S
ADDLICCRQA		S	S	S	S
ADDOBJCRQA		S	S	S	S
ADDPRDCRQA		S	S	S	S
ADDPRDLICI		S	S	S	S
ADDPTFCRQA		S	S	S	S
ADDRSCCRQA		S	S	S	S
APYRMTPTF		S	S	S	S
CHGCMDCRQA		S	S	S	S
CHGCRQA		S	S	S	S
CHGLICCRQA		S	S	S	S
CHGOBJCRQA		S	S	S	S
CHGPRDCRQA		S	S	S	S
CHGPTFCRQA		S	S	S	S
CHGPRDOBJD		S	S	S	S
CHGRSCCRQA		S	S	S	S
CHGSRVPVDA				S	S
CPYPTFSAVF	R	S	S	S	S
CRTPRDDFN		S	S	S	S
CRTPRDLOD		S	S	S	S
CRTPTF		S	S	S	S

Table 10. Authorities of IBM-Supplied User Profiles to Restricted Commands (continued)

Command Name	QSECOFR	QPGMR	QSYSOPR	QSRV	QSRVBAS
CRTPTFPKG		S	S	S	S
DLTPRDDEFN		S	S	S	S
DLTPRDL0D		S	S	S	S
DLTRMTPTF		S	S	S	S
DLTSMGOBJ		S	S	S	S
ENDSYSMGR		S	S	S	S
GENLICKEY		S	S	S	S
HLDPTF		S	S	S	S
INSRMTPRD		S	S	S	S
ORDSPTPTF		S	S	S	S
PKGPRDOPT		S	S	S	S
RLSPTF		S	S	S	S
RMVRMTPTF		S	S	S	S
RSTAPARDTA		S	S	S	S
RTVPRD		S	S	S	S
RTVPTF		S	S	S	S
RTVSMGOBJ		S	S	S	S
RUNSMGCMD		S	S	S	S
RUNSMGOBJ		S	S	S	S
SNDLIC		S	S	S	S
SNDPRD		S	S	S	S
SNDPTF		S	S	S	S
SNDSMGOBJ		S	S	S	S
STRSYSMGR		S	S	S	S
WRKPTF		S	S	S	S
WRKSPTPRD		S	S	S	S
WRKSRVRQS				S	S

Appendix G. Authority required for objects used by commands

The table in this appendix shows what authority is needed for objects referred to by commands.

Table 11. Authority Required for Objects Used by Commands

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
ADDCMDCRQA	Change request description	*CHANGE	*EXECUTE
ADDLICCRQA	Change request description	*CHANGE	*EXECUTE
ADDOBJCRQA	Change request description	*CHANGE	*EXECUTE
ADDPRDCRQA	Change request description	*OBJMGT *CHANGE	*EXECUTE
ADDPTFCRQA	Change request description	*CHANGE	*EXECUTE
ADDRSCCRQA	Change request description	*CHANGE	*EXECUTE
APYRMTPTF	None		
CHGCMDCRQA	Change request description	*CHANGE	*EXECUTE
CHGLICCRQA	Change request description	*CHANGE	*EXECUTE
CHGOBJCRQA	Change request description	*OBJMGT *CHANGE	*EXECUTE
CHGPRDCRQA	Product definition	*CHANGE	*EXECUTE
CHGPRDOBJD	Non-*FILE object	*OBJMGT	*EXECUTE
	*FILE object	*OBJMGT and *OBJOPR	*EXECUTE
CHGPTFCRQA		*CHANGE	*EXECUTE
CHGRSCCRQA		*CHANGE	*EXECUTE
CHGSRVPVDA	None		
CPYPTFSAVF	Tape Device	*USE	
	Optical device	*USE	
	CLRLIB command	*USE	
	DLTLIB command	*USE	
	MOVOBJ command	*USE	
	CRTSAVF command	*USE	
	CHKTAP command	*USE	
CRTPRDDFN	Product definition		*ADD
CRTPRDLOD	Product load		*ADD

Table 11. Authority Required for Objects Used by Commands (continued)

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
CRTPTF	Same as QPZCRTFX API		
DLTPRDDFN	Product definition		
DLTPRDLOD	Product load		
DLTRMTPTF	None		
DLTSMGOBJ		*OBJMGT	
DSPSRVPVDA	None		
ENDSBMCRQA			
ENDSYSMGR			
GENLICKEY			
HLDPTF	None		
ORDSPTPTF	None		
PKGPRDOPT	Folder	*ALL ¹	
RLSPTF	None		
RMVCRQDA	Change request description	*CHANGE	*EXECUTE
RMVRMTPTF	None		
RTVPTF	None		
RTVSMGOBJ		*OBJMGT	
RUNSMGCMD			
RUNSMGOBJ		*OBJMGT	
SBMCRQ	Change request description	*USE	*EXECUTE
SNDLIC	Send License		
SNDPTF	None		
SNDSMGOBJ		*OBJMGT	
WRKCRQD	Change request description	SOME	*EXECUTE
WRKPTF	CRTPTF command	*USE	
	HLDPTF command	*USE	
	DLTPTF command	*USE	
	DSPPTF command	*USE	
	RLSPTF command	*USE	
	CPYPTF command	*USE	
	LODPTF command	*USE	
	SNDPTF command	*USE	
	APYPTF command	*USE	
WRKPRB command	*USE		
WRKSPTPRD	Message file	*USE ²	*EXECUTE ²
WRKSRVRQS	None		

Table 11. Authority Required for Objects Used by Commands (continued)

Command	Referenced Object	Authority Needed for Object	Library Authority (if greater than Read)
Note:			
1	If the product load has folders, the user must be enrolled in the system distribution directory.		
2	If not authorized to the message file for a product, the product description will be blank.		

Appendix H. Authority to System Manager commands and APIs

An authorization list is created for you during the installation process. The authorization list contains System Manager commands that did not exist before you installed the System Manager licensed program. The authorization list (QSM1AUTL) includes all commands and APIs that have *EXCLUDE authority. It can be used to grant users authority to commands and APIs in one single step, instead of having to give individual authority to each command or API. See the Table 12 on page 350 for information about the commands and APIs that are shipped with *EXCLUDE authority.

Notes:

1. If the authorization list is not found during the language installation, the installation continues without the list.
2. If a command is removed from the authorization list, it is only temporarily removed. When the system is upgraded, all commands are added to the list again. This allows you to determine which commands or APIs you want on your list, including any new commands or APIs.
3. The commands are created in the QSYS library, the QSMU library, and the secondary language library.
4. System Manager commands added to the authorization list are the commands from QSMU, the secondary language library, and those copied to QSYS.
5. Adding a user profile to the authorization list only grants *USE authority to the commands and APIs defined in the list. Additional authorities required by specific commands (for example, *JOBCTL special authority) are not automatically given.

A sample program (AUTLPGM) can be found in file QANSSRC in the library QSMU. It is written in CL language and provides an example of how existing users in your system can be added to the QSM1AUTL authorization list. It also shows how an alternate list can be created to handle different levels of authority.

To use this sample in creating your own program, follow these steps:

Notes:

1. Copy the source member to a source physical file member in a different library. You should not change the source in the library QSMU.
2. Tailor the example to your environment by editing the copy source file member.

Note: The sample program performs a check to ensure that it is being run under a user profile having *ALLOBJ or *SECADM special authorities. Removing this check is not recommended.

3. Compile the source using the CRTCLPGM command.
4. Run the compiled program.

If you want different groups of users to have authority to different commands or APIs without using the sample program, follow these steps to create your own authorization list:

- Type CRTAUTL on any command line and press the F4 key.
- Type the name and description of the authorization list.

- Press the F10 key (Additional parameters) to select the authority for the users on the list.
- Press the Enter key to create the list.

The table, Table 12, lists the public authority with which the System Manager commands and their command processing programs are shipped. To use the command, user profiles must have *USE authority to the command definition object in the library QSYS.

Table 12. Public Authority to Commands and APIs

Command or API Name	Processing, Choice, or Prompt Override Program	Shipped Public Authority
ADDCMDCRQA	QNSSDACA ¹	*EXCLUDE *EXECUTE
ADDCRQA	QNSSDACA ¹	*EXCLUDE *EXECUTE
ADDLICCRQA	QNSSDADA ¹	*EXCLUDE *EXECUTE
ADDOBJCRQA	QNSSDADA ¹ QNSSDPOT ²	*EXCLUDE *EXECUTE *USE
ADDPRDCRQA	QNSSDADA ¹	*EXCLUDE *EXECUTE
ADDPRDLICI	QNSLZALI ¹	*EXCLUDE *EXECUTE
ADDPTFCRQA	QNSSDADA ¹	*EXCLUDE *EXECUTE
ADDRSCCRQA	QNSSDARA ¹	*EXCLUDE *EXECUTE
APYRMTPTF	QNSSDPTF ¹ QNSSPPRD ²	*EXCLUDE *EXECUTE *USE
CHGCMDCRQA	QNSSDACA ¹ QNSSDPO ¹	*EXCLUDE *EXECUTE *EXECUTE
CHGCRQA	QNSSDACA ¹ QNSSDPO ³	*EXCLUDE *EXECUTE *USE
CHGLICCRQA	QNSSDADA ¹ QNSSDPO ³	*EXCLUDE *EXECUTE *USE
CHGOBJCRQA	QNSSDADA ¹ QNSSDPO ³ QNSSDPOT ²	*EXCLUDE *EXECUTE *USE *USE
CHGPRDCRQA	QNSSDADA ¹ QNSSDPO ³	*EXCLUDE *EXECUTE *USE
CHGPRDOBJD	QNSSZCPO ¹	*EXCLUDE *EXECUTE

Table 12. Public Authority to Commands and APIs (continued)

Command or API Name	Processing, Choice, or Prompt Override Program	Shipped Public Authority
CHGPTFCRQA	QNSSDADA ¹ QNSSDPO ³	*EXCLUDE *EXECUTE *USE
CHGRSCCRQA	QNSSDARA ¹ QNSSDPO ³	*EXCLUDE *EXECUTE *USE
CHGSRVPVDA	QNSCHGPA ¹ QNSCHGPO ³ QNSLANG ²	*EXCLUDE *EXECUTE *USE *USE
CPYPTFSAVF	QNSPZCPY ¹	*EXCLUDE *EXECUTE
CRTPRDDFN	QNSSZCPD ¹	*EXCLUDE *EXECUTE
CRTPRDLOD	QNSSZCPL ¹	*EXCLUDE *EXECUTE
CRTPTF	QNSPZCRT ¹ QNSPZTGT	*EXCLUDE *EXECUTE *USE
CRTPTFPKG	QNSPZPKG ¹	*EXCLUDE *EXECUTE
DLTPRDDFN	QNSSZDLT ¹	*EXCLUDE *EXECUTE
DLTPRDLOD	QNSSZDLT ¹	*EXCLUDE *EXECUTE
DLTRMTPTF	QNSDSPTF ¹ QNSSPPRD ²	*EXCLUDE *EXECUTE *USE
DLTSBMCRQ	QNSCRDLC ¹	*USE *EXECUTE
DLTSMGOBJ	QNSSDOF ¹ QNSSDPOT ²	*EXCLUDE *EXECUTE *USE
DSPSBMCRQ	QNSCRWKC ¹	*USE *EXECUTE
DSPSBMCRQA	QNSCRWCA ¹	*USE *EXECUTE
DSPSBMCRQM	QNSCRMSG ¹	*USE *EXECUTE
DSPSRVPVDA	QNSDSPPA ¹	*USE *EXECUTE
ENDSBMCRQA	QNSCREA ¹	*USE *EXECUTE
ENDSYSMGR	QNSENDSM ¹	*EXCLUDE *EXECUTE
GENLICKEY	QNSLZGLK ¹	*EXCLUDE *EXECUTE

Table 12. Public Authority to Commands and APIs (continued)

Command or API Name	Processing, Choice, or Prompt Override Program	Shipped Public Authority
HLDPTF	QNSPZHRL ¹	*EXCLUDE *EXECUTE
HLDSBMCRQA	QNSCRHA ¹	*USE *EXECUTE
INSRMTPRD	QNSSDOF ¹	*EXCLUDE *EXECUTE
ORDSPTPTF	QNSORPTF ¹	*EXCLUDE *EXECUTE
PKGPRDOPT	QNSSZPKG ¹	*EXCLUDE *EXECUTE
RLSPTF	QNSPZHRL ¹	*EXCLUDE *EXECUTE
RLSSBMCRQA	QNSCRRA ¹	*USE *EXECUTE
RMVCRQDA	QNSCRRDA ¹	*USE *EXECUTE
RMVRMTPTF	QNSDSPTF ¹ QNSSPPRD ²	*EXCLUDE *EXECUTE *USE
RSTAPARDTA	QNSSCRST ¹	*EXCLUDE *EXECUTE
RTVPRD	QNSSDOF ¹	*EXCLUDE *EXCLUDE
RTVPTF	QNSDSPTF ¹ QNSLANG ² QNSSPPRD ²	*EXCLUDE *EXECUTE *USE *USE
RTVSMGOBJ	QNSSDOF ¹ QNSSDPOT ²	*EXCLUDE *EXECUTE *USE
RUNSMGCMD	QHNSDRCF	*EXCLUDE *EXCLUDE
RUNSMGOBJ	QNSSDOF ¹	*EXCLUDE *EXECUTE
SBMCRQ	QNSCRSBM ¹	*USE *EXECUTE
SNDLIC	QNSSDOF ¹	*EXCLUDE *EXECUTE
SNDPRD	QNSSDOF ¹	*EXCLUDE *EXECUTE
SNDPTF	QNSDSPTF ¹ QNSSPPRD ²	*EXCLUDE *EXECUTE *USE
SNDSMGOBJ	QNSSDOF ¹ QNSSDPOT ²	*EXCLUDE *EXECUTE *USE

Table 12. Public Authority to Commands and APIs (continued)

Command or API Name	Processing, Choice, or Prompt Override Program	Shipped Public Authority
STRYSMGR	QNSSTRSM ¹	*EXCLUDE *EXECUTE
WRKCRQD	QNSCRWKD ¹	*USE *EXECUTE
WRKPTF	QNSPZWRK ¹	*EXCLUDE *EXECUTE
WRKSBMCRQ	QNSCRWKC ¹	*USE *EXECUTE
WRKSBMCRQA	QNSCRWCA ¹	*USE *EXECUTE
WRKSPTPRD	QNSWKSPT ¹	*EXCLUDE *EXECUTE
WRKSRVRS	QNSWRKS ¹	*EXCLUDE *EXECUTE
QNSADDCM		*EXCLUDE
QNSNDCM		*EXCLUDE
QNSUPDST		*EXCLUDE

Note: The superscripted numbers next to the program names denote what type of program it is:

¹ Processing Program

² Choice Program

³ Prompt Override Program

Appendix I. User profile database file

The physical file for the user profile database, QASVNUPP, is copied into the QUSRSYS library when the product is installed. The logical file name is QASVNUP in library QSMU. The files are shipped with the public authority of *EXCLUDE. While the product is being installed, the CCSID of the database files is set to the preferred CCSID of the NLV.

The initial time, last change, and last discovered times are set by the handler's completion exit program when it updates the database on the central site system. All other dates, times, and timestamps are from information gathered at the managed systems. If the network spans several time zones, then the managed system time is used instead of the central site time.

User profile database logical file (QASVNUP)

The user profile database contains information about user profiles in the network.

Table 13. User Profile Database Logical File - QASVNUP

Field	Field Type	Field Description	Field Name
System name	Char(255)	Name of the system where the user profile is located.	UPISYS
System type	Char(1)	The address type of the system name. Possible value is: 1 SNA	UPITYP
User profile name	Char(10)	The user profile name.	UPINAM
Initial discovery date and time	Char(26)	The date and time the user profile was first put in the database. This field is in YYYY-MM-DD-hh.mm.ss.uuuuuu format.	UPIINV
Last changed date and time	Char(26)	The date and time the user profile was last changed. This is updated if there is a change to the entry because of a GUI initiated user profile command or if a change is found in the results of user profile discovery. This field is in YYYY-MM-DD-hh.mm.ss.uuuuuu format.	UPILCH
Last discovery date and time	Char(26)	The date and time the user profile was last discovered by running the user profile discovery change request. This field is in YYYY-MM-DD-hh.mm.ss.uuuuuu format.	UPILDV
Previous sign-on date and time	Char(26)	The date and time the user was last signed on. This field is blank if the user never signed on. This field is in YYYY-MM-DD-hh.mm.ss.uuuuuu format.	UPIPVS
Sign-on attempts not valid	Binary(4)	The number of sign-on attempts that were not valid since the last successful sign-on.	UPINVA
Status	Char(10)	The status of the user profile. Possible values are: *ENABLED The user profile is enabled; therefore, the user is able to sign on. *DISABLED The user profile is disabled; therefore, the user cannot sign on.	UPISTS
Password change date	Char(26)	The date and time the user's password was last changed. This field is in YYYY-MM-DD-hh.mm.ss.uuuuuu format.	UPIPCD

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
No password indicator	Char(1)	If *NONE is specified for the password in the user profile, this field contains a Y. If not, this field is N.	UPINPI
Password expiration interval	Binary(4)	The number of days (from 1 to 366) the user's password can remain active before it must be changed. Special values are: 0 The system value QPWDEXPITV is used to determine the user's password expiration interval. -1 The user's password does not expire (*NOMAX).	UPIPEI
Date password expires	Char(26)	The date and time the user's password will expire. The field is in YYYY-MM-DD-hh.mm.ss.uuuuuu format. If the user's password will not expire (password expiration interval of *NOMAX) or the user's password is set to expire, then this field is blank.	UPIDPE
Days until password expires	Binary(4)	The number of days until the password will expire. This field contains one of the following values: 0 The password is expired. 1 - 7 The number of days until the password expires. -1 The password will not expire in 7 days.	UPIDUE
Set password to expire	Char(1)	Whether the user's password is set to expire, requiring the user to change the password when signing on. Possible values are: Y The user's password is set to expire. N The user's password is not set to expire.	UPISPE
User class name	Char(10)	The field contains one of the following special values: *SECOFR The user has a class of security officer. *SECADM The user has a class of security administrator. *PGMR The user has a class of programmer. *SYSOPR The user has a class of system operator. *USER The user has a class of end user.	UPIUCN

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Special authorities	Char(15)	<p>The special authorities the user has. If the user has the special authority, the field is Y. If not, the field is N. This field contains the following fields:</p> <p>All object Char(1) - Whether the user has all object special authority.</p> <p>Security administrator. Char(1) - Whether the user has security administrator special authority.</p> <p>Job control Char(1) - Whether the user has job control special authority.</p> <p>Spool control Char(1) - Whether the user has spool control special authority.</p> <p>Save system Char(1) - Whether the user has save system special authority.</p> <p>Service Char(1) - Whether the user has service special authority.</p> <p>Audit Char(1) - Whether the user has audit special authority.</p> <p>Input/output system configuration Char(1) - Whether the user has input/output system configuration special authority.</p> <p>Reserved Char(7)</p>	UPISAU
Group profile name	Char(10)	The name of the group profile. If the user does not have a group profile, this field is *NONE.	UPIGPN
Owner	Char(10)	<p>This field indicates who is to own objects created by this user. The field contains one of the following special values:</p> <p>*USRPRF The user owns any objects the user creates. If the user does not have a group profile, the field contains this value.</p> <p>*GRPPRF The user's group profile owns any objects the user creates.</p>	UPIOWN

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Group authority	Char(10)	The authority the user's group profile has to objects the user creates. The possible values are: *NONE The group profile has no authority to the objects the user creates. If the user does not have a group profile, the field contains this value. *ALL The group profile has all authority to the objects the user creates. *USE The group profile has use authority to the objects the user creates. *EXCLUDE The group profile has exclude authority to the objects the user creates.	UPIGAU
Assistance level	Char(10)	The user interface the user will use. Possible values are: *SYSVAL The system value QASTLVL determines which user interface the user is using. *BASIC The user uses the Operational Assist user interface. *INTERMED The user uses the system user interface. *ADVANCED The user uses the expert user interface.	UPIASL
Current library name	Char(10)	The name of the user's current library. If the user does not have a current library, this field is *CRTDFT.	UPICLB
Initial menu name	Char(10)	The initial menu for the user. This field can contain the special value *SIGNOFF.	UPIIMN
Initial menu library name	Char(10)	The name of the library that the initial menu is in. This field can contain the special value *LIBL. If the initial menu name is *SIGNOFF, this field is blank.	UPIIML
Initial program name	Char(10)	The initial program for the user. If the user does not have an initial program, this field is *NONE.	UPIIPN
Initial program library name	Char(10)	The name of the library where the initial program is located. This field can contain the special value *LIBL. If the initial program name is *NONE, this field is blank.	UPIIPL
Limit capabilities	Char(10)	Whether the user has limited capabilities. Possible values are: *PARTIAL The user cannot change his initial program or current library. The user cannot run commands from the command line. *YES The user cannot change his initial menu, initial program, or current library. The user cannot run commands from the command line. *NO The user is not limited.	UPIILCP
Text description	Char(50)	The descriptive text for the user profile.	UPIDSC

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Display sign-on information	Char(10)	Whether the sign-on information display is shown when the user signs on. Possible values are: *SYSVAL The system value QDSPSGNINF determines if the sign-on information display is shown when the user signs on. *YES The sign-on information display is shown when the user signs on. *NO The sign-on information display is not shown when the user signs on.	UPIDSI
Limit device sessions	Char(10)	Whether the user is limited to one device session. Possible values are: *SYSVAL The system value QLMTDEVSSN determines whether the user is limited to one device session. *YES The user is limited to one device session. *NO The user is not limited to one device session.	UPIILDS
Keyboard buffering	Char(10)	This field indicates the keyboard buffering value that is used when a job is initialized for this user. Possible values are: *SYSVAL The system value QKBDBUF determines the keyboard buffering value for this user. *YES The type-ahead and attention-key buffering options are both on. *NO The type-ahead and attention-key buffering options are not on. *TYPEAHEAD The type-ahead option is on, but the attention-key buffering option is not.	UPIKYB
Maximum allowed storage	Binary(4)	The maximum amount of auxiliary storage (in kilobytes) that can be assigned to store permanent objects owned by the user. If the user does not have a maximum amount of allowed storage, this field contains -1 for *NOMAX.	UPIMAX
Storage used	Binary(4)	The amount of auxiliary storage (in kilobytes) occupied by this user's owned objects.	UPIUSE
Highest scheduling priority	Char(1)	The highest scheduling priority the user is allowed to have for each job submitted to the system. The priority is a value from 0 through 9, with 0 being the highest priority.	UPISCD
Job description name	Char(10)	The name of the job description used for jobs that start through subsystem workstation entries.	UPIJDN
Job description name library	Char(10)	The name of the library where the job description is located. This field can contain the special value *LIBL.	UPIJDL
Accounting code	Char(15)	The accounting code associated with this user. If the user does not have an accounting code, this field is blank.	UPIACC

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Message queue name	Char(10)	The name of the message queue that is used by this user.	UPIMQN
Message queue library name	Char(10)	The name of the library where the message queue is located. This field can contain the special value *LIBL.	UPIMQL
Message queue delivery method	Char(10)	How the messages are delivered to the message queue used by the user. Possible values are: *BREAK The job to which the message queue is assigned is interrupted when a message arrives on the message queue. *DFT Messages requiring replies are answered with their default reply. *HOLD The messages are held in the message queue until they are requested by the user or program. *NOTIFY The job to which the message queue is assigned is notified when a message arrives on the message queue.	UPIMQD
Message queue severity	Binary(4)	The lowest severity that a message can have and still be delivered to a user in break or notify mode. The severity is a value from 0 through 99.	UPIMQS
Output queue	Char(10)	The output queue used by this user. This field can contain one of the following special values: *WRKSTN The output queue assigned to the user's workstation is used. *DEV An output queue with the same name as the device specified in the printer device parameter is used.	UPIOQN
Output queue library	Char(10)	The name of the library where the output queue is located. This field can contain the special value *LIBL. If the output queue is *WRKSTN or *DEV, this field is blank.	UPIOQL
Print device	Char(10)	The printer used to print for this user. Possible values are: *WRKSTN The printer assigned to the user's work station is used. *SYSVAL The default system printer specified in the system value QPRTDEV is used.	UPIPRT

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Special environment	Char(10)	The special environment the user operated in after signing on. Possible values are: *SYSVAL The system value QSPCENV determines the user's special environment. *NONE The user operates in the OS/400 environment. *S36 The user operates in the System/36™ environment.	UPISEN
Attention-key-handling program name	Char(10)	The Attention-key-handling program for this user. This field may contain one of the following special values: *SYSVAL The system value QATNPGM determines the user's Attention-key handling program. *NONE No Attention-key-handling program is used by this user. *ASSIST The Operational Assistant Attention-Key-Handling API (QEZMAIN) is used by the user.	UPIAHP
Attention-key-handling program library	Char(10)	The name of the library where the Attention-key-handling program is located. This field can contain the special value *LIBL. If the program name is a special value, this field is blank.	UPIAHL
Language ID	Char(10)	The language ID used by the system for this user. The field can contain the following special value: *SYSVAL The system value QLANGID determines the user's language ID.	UPILAN
Country or Region ID	Char(10)	The country or region ID used by the system for this user. The field can contain the following special value: *SYSVAL The system value QCNTYID determines the user's country or region ID.	UPICTY
Character code set ID	Binary(4)	The character code set used by the system for this user. The field can contain the following special value: -2 The system value QCCSID determines the user's character code set.	UPISID

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
User options	Char(36)	<p>The options for users to customize their environment. This field contains the following fields:</p> <p>Char(1) Show keywords (*CLKWD). Whether the keywords are shown when a CL command is displayed. If the keywords are to be shown, this field is Y. If not, this field is N.</p> <p>Char(1) Show detailed information (*EXPERT). Whether more detailed information is shown when the user is defining or changing the system using edit or display object authority. This user option is independent of the ASTLVL parameter on the user profile and the ASTLVL parameter available on commands. If the details are to be shown, this field is Y. If not, this field is N.</p> <p>Char(1) Full screen help (*HLPFULL). Whether UIM online help is to be displayed on a full screen or in a window. If the full screen is to be shown, this field is Y. If not, this field is N.</p> <p>Char(1) Show status messages (*STSMSG). Whether status messages sent to the user are shown. If the status messages are to be shown, this field is Y. If not, this field is N.</p> <p>Char(1) Do not show status messages (*NOSTSMSG). Whether status messages sent to the user are not shown. If the status messages are not to be shown, this field is Y. If they are to be shown, this field is N.</p> <p>Char(1) Roll key direction change (*ROLLKEY). Whether the opposite action from the system default for roll keys is taken or not. If the opposite action is to be taken, this field is Y. If not, this field is N.</p> <p>Char(1) Printing complete message (*PRTMSG). Whether a message is sent to the user when a spooled file is printed or not. If a message is to be sent, this field is Y. If not, this field is N.</p> <p>Char(29) Reserved.</p>	UPIUSO

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Sort sequence table name	Char(10)	<p>The name of the sort sequence table used for string comparisons. The following special values can be returned:</p> <p>*HEX The hexadecimal values of the characters are used to determine the sort sequence.</p> <p>*LANGIDUNQ The unique-weight sort table associated with the language specified is used to determine the sort sequence.</p> <p>*LANGIDSHR The shared-weight sort table associated with the language specified is used to determine the sort sequence.</p> <p>*SYSVAL The system value QSRTSEQ is used to determine the sort sequence.</p>	UPI SST
Sort sequence table library name	Char(10)	The name of the library where the sort sequence table is located. If a special value is used, this field is blank.	UPI SSL
Object audit value	Char(10)	<p>The current user's object auditing value. Possible values are:</p> <p>*NONE No additional object auditing is done for the current user.</p> <p>*CHANGE Object changes are audited for the current user if the object's auditing value is *USRPRF.</p> <p>*ALL Object read and change operations are audited for the current user if the object's auditing value is *USRPRF.</p>	UPI OAV

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
User action audit level	Char(64)	<p>The action audit values for this user. If the user has a specific audit value, the field is Y. If not, the field is N. This field contains the following:</p> <p>*CMD Char(1) The user has the *CMD audit value specified in the user profile.</p> <p>*CREATE Char(1) The user has the *CREATE audit value specified in the user profile.</p> <p>*DELETE Char(1) The user has the *DELETE audit value specified in the user profile.</p> <p>*JOBDTA Char(1) The user has the *JOBDTA audit value specified in the user profile.</p> <p>*OBJMGT Char(1) The user has the *OBJMGT audit value specified in the user profile.</p> <p>*OFCSRVR Char(1) The user has the *OFCSRVR audit value specified in the user profile.</p> <p>*PGMADP Char(1) The user has the *PGMADP audit value specified in the user profile.</p> <p>*SAVRST Char(1) The user has the *SAVRST audit value specified in the user profile.</p> <p>*SECURITY Char(1) The user has the *SECURITY audit value specified in the user profile.</p> <p>*SERVICE Char(1) The user has the *SERVICE audit value specified in the user profile.</p> <p>*SPLFDTA Char(1) The user has the *SPLFDTA audit value specified in the user profile.</p> <p>*SYSMGT Char(1) The user has the *SYSMGT audit value specified in the user profile.</p> <p>Char(29) Reserved</p>	UPIUAL

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Group authority type	Char(10)	<p>The type of authority the user's group profile has to objects the user creates. Possible values are:</p> <p>*PRIVATE The group profile has a private authority to the objects the user creates. If the user does not have a group profile, the field contains this value.</p> <p>*PRIGRP The group profile will be the primary group for objects the user creates.</p>	UPIGAT
User ID number	Binary(4)	The user ID (uid) number for the user profile. The uid is used to identify the user when it is using the directory file system.	UPIIDN

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Locale job attributes	Char (16)	<p>The job attributes that are taken from the user's locale path name. If a particular job attribute is taken from the locale path name, the specific field is Y (yes). If not, the specific field is N (no). The possible values are:</p> <p>*NONE CHAR(1) No job attributes are used from the locale path name at the time a job is started for this user profile.</p> <p>*SYSVAL CHAR(1) The job attributes assigned from the locale path name are determined by the system value QSETJOBATR at the time a job is started for this user profile.</p> <p>*CCSID CHAR(1) The coded character set identifier is set from the locale path name at the time a job is started for this user profile.</p> <p>*DATFMT CHAR(1) The date format is set from the locale path name at the time a job is started for this user profile.</p> <p>*DATSEP CHAR(1) The date separator is set from the locale path name at the time a job is started for this user profile.</p> <p>*SRTSEQ CHAR(1) The sort sequence is set from the locale path name at the time a job is started for this user profile.</p> <p>*TIMSEP CHAR(1) The time separator is set from the locale path name at the time a job is started for this user profile.</p> <p>Reserved CHAR(9) An ignored field.</p>	UPISJA
Group ID number	Binary(4)	The group ID number for the user profile. The group ID number is used to identify the user when it is a group and a member of the group is using the directory file system.	UPIDNM

Table 13. User Profile Database Logical File - QASVNUP (continued)

Field	Field Type	Field Description	Field Name
Supplemental groups	Array(*) of Char(10)	The array of supplemental groups for the user profile.	UPIGRP
Home directory	Char(*)	<p>The home directory for this user profile. The home directory is the user's initial working directory. The working directory, associated with a process, is used in path name resolution in the directory file system for path names that do not begin with a slash (/). The structure for the home directory name returned is:</p> <p>Binary(4) CCSID of the returned home directory name</p> <p>Char(2) Country or Region ID</p> <p>Char(3) Language ID</p> <p>Char(3) Reserved</p> <p>Binary(4) Flag byte</p> <p>Binary(4) Number of bytes in the home directory name</p> <p>Char(2) Home directory delimiter</p> <p>Char(10) Reserved</p> <p>Char(*) Home directory name value</p>	UPIDIR
Locale	Char (*)	An object that contains information about valid alphabetic characters, the collating sequence, the format of numbers and currency amounts, and the format of date and time. This information is used by certain C/400 [®] library functions in application programs. The system-recognized identifier for the object type, C locale description, is *CLD.	UPILOC

Appendix J. QSVSM user profile default values

These are the shipped or default authorities for the QSMU user profile.

One way to restore the original (shipped) values of QSVSM user profile is by checking current values or authorities for this profile and, if needed, reset them to the original values using Change User Profile (CHGUSRPRF), Edit Object Authority (EDTOBJAUT), Grant Object Authority (GRTOBJAUT), or Revoke Object Authority (RVKOBJAUT) commands.

```

/*
QSYS/CRTUSRPRF USRPRF(QSVSM) PASSWORD(*NONE) USRCLS(*SYSOPR) +
STATUS(*DISABLED) PWDEXP(*NO) INLPGM(*NONE) +
ASTLVL(*SYSVAL) CURLIB(*CRTDFT) LMTCPB(*NO) +
INLMNU(*LIBL/MAIN) SPCAUT(*JOBCTL) +
SPCENV(*SYSVAL) +
DSPSGNINF(*SYSVAL) PWDEXPITV(*SYSVAL) +
LMTDEVSSN(*SYSVAL) KBDBUF(*SYSVAL) MAXSTG(*NOMAX) +
PTYLMT(3) JOBDB(*LIBL/QDFTJOBDB) GRPPRF(*NONE) +
OWNER(*USRPRF) GRPAUT(*NONE) ACGCDE(*BLANK) +
DOCPWD(*NONE) MSGQ(*USRPRF) DLVRY(*NOTIFY) SEV(0) +
PRTDEV(*WRKSTN) OUTQ(*WRKSTN) ATNPGM(*SYSVAL) +
LANGID(*SYSVAL) CNTRYID(*SYSVAL) CCSID(*SYSVAL) +
USROPT(*NONE) AUT(*EXCLUDE) +
TEXT('SystemView System Manager User Profile')

QSYS/CHGOBJOWN OBJ(QSVSM) +
OBJTYPE(*USRPRF) +
NEWOWN(QSYS) +
CUROWNAUT(*REVOKE)

QSYS/GRTOBJAUT OBJ(QGPL/QAPZCOVER) +
OBJTYPE(*FILE) +
USER(QSVSM) +
AUT(*USE)

QSYS/GRTOBJAUT OBJ(QSYS/QSYSNOMAX) +
OBJTYPE(*JOBQ) +
USER(QSVSM) +
AUT(*USE)

QSYS/GRTOBJAUT OBJ(QSMU/QNSCRJRN) +
OBJTYPE(*JRN) +
USER(QSVSM) +
AUT(*ALL)

QSYS/GRTOBJAUT OBJ(QSMU/QANSCRAC) +
OBJTYPE(*FILE) +
USER(QSVSM) +
AUT(*ALL)

QSYS/GRTOBJAUT OBJ(QSMU/QANSCRAN) +
OBJTYPE(*FILE) +
USER(QSVSM) +
AUT(*ALL)

QSYS/GRTOBJAUT OBJ(QSMU/QANSCRCN) +
OBJTYPE(*FILE) +
USER(QSVSM) +
AUT(*ALL)

```

QSYS/GRTOBJAUT	OBJ(QSMU/QANSCRCR)	+
	OBJTYPE(*FILE)	+
	USER(QSVSM)	+
	AUT(*ALL)	
QSYS/GRTOBJAUT	OBJ(QSMU/QANSCRMS)	+
	OBJTYPE(*FILE)	+
	USER(QSVSM)	+
	AUT(*ALL)	
QSYS/GRTOBJAUT	OBJ(QSYS/CPYPTF)	+
	OBJTYPE(*CMD)	+
	USER(QSVSM)	+
	AUT(*USE)	

Bibliography

Bibliography

The following lists systems management tasks with the publications or Information Center topics that describe them.

Systems Management

This list is provided to help you find more information about setting up communications and using the other OS/400 functions related to problem management and change management.

For more information about installing licensed programs and new releases, see:

- *Software Installation*, SC41-5120-06
- *Managed System Services for AS/400 Use*, SC41-3323-02

Software Management

| For more information about maintaining OS/400
| software and using individual PTFs and
| cumulative PTF packages, see the Software and
| licensed program topic under the Systems
| Management category in the Information Center.

Communications and Network Planning

For more information about communications and network planning, see:

- | • Central Site Distribution topic under the
| Systems Management category in the
| Information Center.
- | • *Alerts Support*, SC41-5413-00
- | • ISDN topic under the Networking category in
| the Information Center.
- | • *LAN, Frame-Relay and ATM Support*,
| SC41-5404-01
- | • Frame relay topic under the Networking
| category in the Information Center.
- | • ATM topic under the Networking category in
| the Information Center.
- | • *Communications Management*, SC41-5406-02
- | • *X.25 Network Support*, SC41-5405-01
- | • *Communications Configuration*, SC41-5401-00

- | • *TCP/IP Configuration and Reference*, SC41-5420-04
| and the TCP/IP topic under the Networking
| category in the Information Center.
- | • *Simple Network Management Protocol (SNMP)*
| *Support*, SC41-5412-00
- | • APPC, APPN, and HPR topic under the
| Networking category in the Information Center.

For more information about planning for remote support, see *Software Installation*, SC41-5120-06

| For more information about how to set up your
| initial work management environment and
| change system objects to meet your needs, see
| *Work Management*, SC41-5306-03 (V4R4) and the
| Work Management topic under the Systems
| Management category in the Information Center.

System Operation

For more information about system operations tasks such as working with the system operator functions; using copy screen image (STRCPYSCN); using customer support operations; running problem analysis; ordering, applying, and distributing PTFs; working with IBM product information; or using the technical information exchange, see:

- | • Software and licensed programs topic under
| the Systems Management category in the
| Information Center.
- | • Basic system operations topic under the
| Systems Management category in the
| Information Center.

National Language Support

| For more information about planning for and
| using national language support functions, see the
| Globalization topic under the Programming
| ->Programming support category in the
| Information Center.

iSeries Access information

| For more information about iSeries Access, see the
| iSeries Access topic under the Connecting to the
| iSeries category in the Information Center.

NetView Distribution Information

For more information about NetView Distribution Manager, see:

- *Network Licensed Programs NetView Distribution Manager General Information*, GH19-6792
- *Network Licensed Programs NetView Distribution Manager Installation and Customization*, SH19-6794
- *Network Licensed Programs NetView Distribution Manager Messages and Codes*, SH19-6798
- *Network Licensed Programs NetView Distribution Manager User's Guide*, SH19-6795

Backup and Recovery

For more information about backup and recovery and application program interfaces (APIs), see:

- *Backup and Recovery*, SC41-5304-06
- | • Backup and Recovery topic under the Systems Management category in the Information Center. This topic contains information on how to plan a backup and recovery strategy, how to set up disk protection for your data, how to back up your system, and how to control your system shutdown in the event of a failure.
- | • API topic under the Programming category in the Information Center.

Security Information

For more information about security issues, see the *iSeries Security Reference*, SC41-5302-06.

Index

Special Characters

- *IBMSRV (IBM service support) 8
- *TEST parameter
 - Send Service Request (SNDSRVRQS) command 30
- *UNATTEND network attribute
 - value 11

A

- active job
 - working with 260
- activity
 - adding to change request
 - description 199
 - change request description 197
 - conditioning 202
 - details
 - displaying 205
 - problems 208
 - scheduling 201
 - types, defining 204
 - what if it doesn't run 208
- add activity exit program 308
- Add Change Management Activity (QNSADDCM) API 292
- Add Product License Information (ADDPRDLICI) command 93, 96, 98
 - authorized IBM-supplied user profiles 343
- adding
 - activities to a change request
 - description 199
 - directory entries
 - on the service provider 236
 - on the service requester 236
 - service provider example 246
 - service requester example 251
 - distribution queues 235, 242, 249
 - entries
 - directory 28
 - service provider 250
 - service requester 19, 20, 244
 - example
 - directory entries on service provider 246
 - distribution queues on the service provider 242
 - distribution queues on the service requester 249
 - product support 262
 - routing table entries on the service requester 249
 - service provider entries 250
 - information to PTF database 25
 - language feature support
 - service provider 24
 - license information 98
 - problem description 162

- adding (*continued*)
 - problem records
 - problem analysis results 168
 - product support
 - example 262
 - licensed program offerings 23
 - performance 23
 - programming requests for price quotations (PRPQS) 23
 - service provider 23
 - PTF information to problem record 190
 - routing table entries
 - on the service provider 236, 243
 - on the service requester 236, 249
 - service provider entries 27
 - sphere of control 12
 - symptom strings to the PTF 145
- ADDPRDCRQA (Add Product Change Request Activity) command
 - object authority required 345
- ADDPRDLICI (Add Product License Information) command 93, 96, 98
 - authorized IBM-supplied user profiles 343
- Advanced Peer-to-Peer Networking (APN) 8
 - communications mode #INTER 8
 - control point 237
 - definition 7
 - network ID 237
 - requirements 7
- alert controller description (ALRCTLTD)
 - attribute
 - changing the description 12
 - example 33
- alert log
 - cleaning up disk space 12
- alert logging status (ALRLOGSTS)
 - attribute
 - example 258
 - setting 11
- alert primary focal point (ALRPRIFP)
 - attribute 11
- alert status (ALRSTS) attribute
 - example 258, 266
 - setting 11
- alerts
 - alert focal point example 31
 - changing
 - alert controller description (ALRCTLTD) 12
 - definition 10
 - example 257
 - filtering 13
 - focal point definition 10
 - grouping 13
 - information in 11
 - problem notification 183
 - problem notifications 10
 - creating problem record 153

- alerts (*continued*)
 - problem notifications (*continued*)
 - system-detected problems 10
 - problem records not created 272
 - setting up basic support 11
 - sphere of control 11
 - using 10
 - working with 12, 185, 270
- ALRCTLTD (alert controller description)
 - attribute
 - changing the description 12
 - example 33
- ALRLOGSTS (alert logging status)
 - attribute
 - example 258
 - setting 11
- ALRPRIFP (alert primary focal point)
 - attribute 11
- ALRSTS (alert status) attribute
 - example 258, 266
 - setting 11
- Analyze Problem (ANZPRB)
 - command 150
 - authorization to 29, 275
 - collecting problem data 181
 - creating a problem record 153, 183, 186
 - description 194
 - example 261, 267
 - example performance problems 261
 - problem status changes 175
 - problems 150
 - starting remote problem analysis 186
- analyzing
 - example problems
 - communications 260
 - local problems
 - change problem status back to prepared 185
 - system-detected 164, 184
 - user-detected 181
 - problems
 - example performance problem 266
 - remote
 - introduction 179
 - remote problems
 - from command line 186
 - from message queue 185
 - from problem log 185, 187
 - from Work with Alerts
 - display 185, 186
 - preparing for 184
 - system-detected problems 184
 - user-detected problems 186
- answering problems
 - changing status to ANSWERED 171
 - hardware 165, 192
 - information 166
 - introduction 188
 - list of possible causes 166

- answering problems (*continued*)
 - list of PTFs 166
 - searching for PTFs 165
 - software 165, 189
- ANZPRB (Analyze Problem) command
- authorization to 29, 275
- collecting problem data 181
- creating a problem record 153
- example 261, 267
- example performance problems 261
- problem status changes 175
- problems 150
- starting remote problem analysis 186
- APAR (authorized program analysis report)
 - creating 181
 - definition 166
 - deleting 167
 - displaying 167
 - saving 167
- API
 - Change Management Activity 292
 - Change request 292
 - QLZAREQ (Request License) 103
 - QLZARLS (Release License) 103
 - QLZARTV (Retrieve License Information) 103
 - QNSSNDCM (send change request message) 303
 - QNSUPDST (update status) 306
 - Release License (QLZARLS) 103
 - Request License (QLZAREQ) 103
 - Retrieve License Information (QLZARTV) 103
 - Send Change Request Message (QNSSNDCM) 303
 - Update Status (QNSUPDST) 306
- application environment for complex products 39
- application program interface (API)
 - authority considerations for license management APIs 105
 - calls for license management 102
 - change request 292
 - problem management 195
 - QNSADDCM 292
 - release license (QLZARLS) 102, 103
 - request license (QLZAREQ) 102, 103
 - retrieve license information (QLZARTV) 102
 - Send Change Request Message (QNSSNDCM) 303
 - update status 306
- Apply Program Temporary Fix (APYPTF) command 274
- applying
 - program temporary fix 274
- applying PTFs
 - corequisites 274
 - prerequisites 274
- APPN mode 8
- APYPTF (Apply Program Temporary Fix) command 274
- APYPTF command 148
- attribute
 - CHGNETA (Change Network Attributes) 241

- attribute (*continued*)
 - example
 - changing network 241, 248
 - network
 - alert controller description (ALRCTLD) 12, 33
 - alert logging status (ALRLOGSTS) 11
 - alert primary focal point (ALRPRIFP) 11
 - alert status (ALRSTS) 11
 - alert status *UNATTEND 11
 - changing 11
 - changing alert controller description (ALRCTLD) 12
 - example 241, 257, 269
 - server (NETSERVER) 33
 - setting 11
 - setting up alerts 32
 - verifying system name 10
 - running remote
 - unattended 11
 - service provider 21, 257, 263
 - changing 18, 33
 - definitions 18
 - determining default message queue 121
 - example 257
 - hold PTFs received (HLDPTF) 131
 - log remote problems (LOGRMTPRB) 18, 153, 272
 - monitoring the message queue (MSGQ) 263
 - not set correctly 272, 273
 - releasing PTFs 132
 - send PTFs automatically (SNDPTFAUTO) 22, 123, 273
 - setting 18, 153
 - service requester
 - creating problem record 182
 - log remote problems (LOGRMTPRB) 182
 - service requester information 21
 - authority
 - commands 343
 - granting 17
 - processing 349
 - authority considerations for license management APIs 105
 - authorization list
 - creating for service provider commands 17
 - working with 17
 - authorized program analysis report (APAR)
 - creating 181
 - definition 166
 - deleting 167
 - displaying 167
 - saving 167
 - automatic cleanup functions
 - reclaiming disk space
 - from deleted alerts 12
 - from deleted problems 163
 - reclaiming storage
 - from deleted PTF save files 132

- automatic virtual device configuration (QAUTOVRT) system value
 - changing 30
 - definition 29
 - example 253
 - not set correctly 275

B

- backup and recovery
 - procedures 149
 - product 113
 - PTF 149
- basic security representative (QSRVBAS)
 - receiving files 123
- basic service representative (QSRVBAS)
 - user profile 236
- books
 - list 371

C

- CFGDSTSRV (Configure Distribution Services) command
 - correcting SNADS problems 274
 - SNADS setup example 242, 249
 - using to set up SNADS 235
- change activity exit program 310
- change management
 - overview 115
 - using System Manager functions
 - with central site systems 197
 - Work with Program Temporary Fix (WRKPTF) command 127
- Change Network Attributes (CHGNETA) command
 - example
 - service provider 240
 - service requester 248
 - with a S/370 host 35
 - setting up alerts 11, 33
- Change Problem (CHGPRB) command
 - changing text in fields 161
 - example 162
- Change Product Object Description (CHGPRDOBJD) command
 - authorized IBM-supplied user profiles 343
 - object authority required 345
- change request
 - report capabilities 210
 - submitting 208
 - tracking submitted 210
- change request description
 - activities 197
 - adding activities 199
 - considerations 209
 - creating 198
 - definition 197
- Change Service Provider Attributes (CHGSRVPVDA) command
 - authorized IBM-supplied user profiles 343
 - correcting attribute values 272, 273
 - example network 33
 - object authority required 345

Change Service Provider Attributes (CHGSRVPVDA) command (*continued*)
 setting up 18

Change System Value (CHGSYSVAL) command
 correcting values 275
 service requester example 253
 using 30, 163

changing 272
 alert controller description (ALRCTLD) attribute 12
 change request description 199
 correcting values 275
 entries
 service provider 28
 service requester 20
 sphere of control 12
 group assigned parameter 162
 network attributes 11, 35, 240
 alert controller description (ALRCTLD) 12
 example 33
 service provider 241
 setting up alerts 11
 problem 161
 description 162
 severity 161
 problem status
 ANSWERED 171
 CLOSED 173
 description 167
 for local analysis 186
 OPENED 167
 PREPARED 169
 READY 168
 SENT 170
 VERIFIED 172
 route to IBM service support (*IBMSRV) 29, 251
 service number parameter 162
 service provider attributes 18
 system value 253, 275
 QAUTOVRT (automatic virtual device configuration) 30, 253
 QPRBHLDITV (problem hold interval) 163
 QRMTSIGN (remote sign-on verification) 30, 253
 user-assigned parameter 162

changing information
 problem
 record text 161

Check Product Option (CHKPRDOPT) command 76

checking
 a product using an exit program 76
 change requests 210
 job log 274
 product option 76
 PTF requisites 122

CHGNETA (Change Network Attributes) command
 example
 service provider 240
 service requester 248
 with a S/370 host 35
 setting up alerts 11, 33

CHGPRB (Change Problem) command
 changing text in fields 161
 displays 161
 example 162

CHGPRDOBJD (Change Product Object Description) command
 authorized IBM-supplied user profiles 343
 object authority required 345

CHGSRVPVDA (Change Service Provider Attributes) command
 authorized IBM-supplied user profiles 343
 correcting attribute values 272
 displays 18
 example network 33
 object authority required 345
 setting up 18

CHGSRVPVDA (Change Service Provider Attributes) Command
 correcting attribute values 273

CHGSYSVAL (Change System Value) command 275
 service requester example 253
 using 30, 163

CHKPRDOPT (Check Product Option) command 76
 cleaning up
 alert log file members 12
 problem log file members 163
 PTF save files and file members 132

cleanup functions
 reclaiming disk space
 from deleted alerts 12
 from deleted problems 163
 from deleted PTF save files and information 132

closing
 problem records 166

code load 52
 used to define features 96

collecting
 information
 about PTFs 129
 Query 129
 system-detected problems 180
 user-detected problems 181
 problem records 166

command
 Analyze Problem (ANZPRB) authorization to 29
 Change Network Attributes (CHGNETA) 11
 example with a S/370 host 35
 setting up alerts 11, 33
 Change Service Provider Attributes (CHGSRVPVDA) 18
 example network 33
 Change System Value (CHGSYSVAL) using 30
 CHGNETA (Change Network Attributes) setting up alerts 11, 33
 CHGSRVPVDA (Change Service Provider Attributes) 18
 CHGSYSVAL (Change System Value) using 30

command (*continued*)
 Create User Profile (CRTUSRPRF) for problem management tasks 29
 CRTUSRPRF (Create User Profile) for problem management tasks 29
 Display Network Attributes (DSPNETA) using to verify system name 10
 Display Service Provider Attributes (DSPSRVPVDA) description 19
 Display System Value (DSPSYSVAL) example 21
 displays 18
 DSPNETA (Display Network Attributes) using to verify system name 10
 DSPSRVPVDA (Display Service Provider Attributes) 19
 DSPSYSVAL (Display System Value) example 21
 example 33
 Grant Object Authority (GRTOBJAUT) setting up command authority 17
 GRTOBJAUT (Grant Object Authority) setting up command authority 17
 map for setting up 6
 Print Error Log (PRTERLOG) 29
 authorization to 29
 PRTERLOG (Print Error Log) 29
 authorization to 29
 Save Object (SAVOBJ) 25
 description 25
 SAVOBJ (Save Object) description 25
 Send Service Request (SNDSRVRQS) testing communications configuration 30
 setting up 18
 setting up alerts 11
 SNDSRVRQS (Send Service Request) testing communications configuration 30
 Work with Alerts (WRKALR) displaying alerts 12
 Work with Authorization Lists (WRKAUTL) 17
 Work with Configuration List (WRKCFGL) 18
 Work with Contact Information (WRKCNTINF) verifying service information 14
 Work with Controller Descriptions (WRKCTLD) 12
 Work with Problem (WRKPRB) authorization to 29
 Work with Service Providers (WRKSRVPVD) adding entries 27
 changing route to IBM service support 34
 Work with Service Requesters (WRKSRVRQS) adding entries 19
 example 33
 Work with Sphere of Control (WRKSOC) 12, 35

command (*continued*)
 using 12
 Work with Supported Products (WRKSPTPRD)
 description 22
 example 33
 WRKCFGL (Work with Configuration List) 18
 WRKCNTINF (Work with Contact Information)
 verifying service information 14
 WRKCTLD (Work with Controller Descriptions) 12
 WRKPRB (Work with Problem)
 authorization to 29
 WRKSOC (Work with Sphere of Control)
 using 12
 WRKSPTPRD (Work with Supported Products)
 description 22
 example 33
 WRKSRVPVD (Work with Service Providers)
 adding entries 27
 changing route to IBM service support 34
 WRKSRVRQS (Work with Service Requesters) 33
 command reference 233
 command, CL 115, 186, 263
See also individual commands for authority restrictions
See also object authority for authority restrictions
 adding product support 275
 Analyze Problem (ANZPRB) 150
 authorization to 275
 collecting problem data 181
 creating a problem record 153, 183, 186
 description 194
 example 261, 267
 problem status changes 175
 starting remote problem analysis 186
 ANZPRB (Analyze Problem)
 authorization to 275
 description 194
 Apply Program Temporary Fix (APYPTF) 274
 APYPTF 148
 authority 343
 authorized IBM-supplied user profiles 343
 CFGDSTSRV (Configure Distribution Services)
 SNADS setup example 242, 249
 using to set up SNADS 235
 Change Network Attributes (CHGNETA)
 example service provider 240
 service requester example 248
 Change Problem (CHGPRB)
 changing text in fields 161
 example 162

command, CL (*continued*)
 Change Service Provider Attributes (CHGSRVPVDA) 272
 correcting attribute values 272, 273
 Change System Value (CHGSYSVAL)
 correcting values 275
 service requester example 253
 using 163
 CHGNETA (Change Network Attributes)
 example service provider 240
 service requester example 248
 CHGPRB (Change Problem)
 changing text in fields 161
 example 162
 CHGSRVPVDA (Change Service Provider Attributes) 273
 CHGSYSVAL (Change System Value)
 service requester example 253
 using 163
 Configure Distribution Services (CFGDSTSRV) 249
 correcting SNADS problems 274
 SNADS setup example 242
 using to set up SNADS 235
 Copy Program Temporary Fix (CPYPTF)
 restrictions 125
 Copy PTF to Save File (CPYPTFSAVF)
 description 123
 receiving PTFs 124
 CPYPTF (Copy Program Temporary Fix)
 restrictions 125
 CPYPTFSAVF (Copy PTF to Save File)
 description 123
 receiving PTFs 124
 Create Controller Description (APPC) (CRTCTLAPPC) 248
 example 241
 Create Line Description (Token-Ring) (CRTLINTRN)
 example 241, 248
 Create Message File (CRTMSGF) 49
 Create Product Definition (CRTPRDDFN)
 authorized IBM-supplied user profiles 343
 Create Product Load (CRTPRDL0D)
 authorized IBM-supplied user profiles 343
 using multiple libraries 55
 Create Program Temporary Fix (CRTPTF) 343
 using different release levels 142
 Create Program Temporary Fix Package (CRTPTFPKG) 136, 149
 authorized IBM-supplied user profiles 344
 Create User Profile (CRTUSRPRF)
 example 252
 CRTCTLAPPC (Create Controller Description (APPC)) 248
 example 241

command, CL (*continued*)
 CRTLINTRN (Create Line Description (Token-Ring))
 example 241, 248
 CRTMSGF (Create Message File) 49
 CRTPRDDFN (Create Product Definition)
 authorized IBM-supplied user profiles 343
 CRTPRDL0D (Create Product Load)
 authorized IBM-supplied user profiles 343
 using multiple libraries 55
 CRTPTF (Create Program Temporary Fix)
 authorized IBM-supplied user profiles 343
 using different release levels 142
 CRTPTFPKG (Create Program Temporary Fix Package) 136, 149
 authorized IBM-supplied user profiles 344
 CRTUSRPRF (Create User Profile)
 example 252
 Delete Problem (DLTPRB) 162
 Delete Program Temporary Fix (DLTPTF)
 description 132
 save file management 125
 description 150
 Display Job Log (DSPJOBLOG)
 applying PTF 274
 checking job log 274
 example 266
 Display Messages (DSPMSG)
 displaying service provider message queue 185
 example 270
 Display Problem (DSPPRB) 159
 Display Program Temporary Fix (DSPPTF)
 checking PTF status 124, 273
 description 128
 example 129, 261
 verifying PTF status 274
 Display Service Provider Attributes (DSPSRVPVDA)
 determining default message queue 121, 272, 273
 object authority required 345
 verifying attribute values 272, 273
 Display System Value (DSPSYSVAL)
 verifying the current value 275
 DLTPRB (Delete Problem) 162
 DLTPTF (Delete Program Temporary Fix)
 description 132
 save file management 125
 DSPJOBLOG (Display Job Log)
 applying PTF 274
 checking job log 274
 example 266
 DSPMSG (Display Messages)
 displaying service provider message queue 185
 example 270
 DSPPRB (Display Problem) 159

- command, CL (*continued*)
 - DSPPTF 148
 - DSPPTF (Display Program Temporary Fix)
 - checking PTF status 124
 - example 129
 - verifying PTF status 274
 - DSPSRVPVDA (Display Service Provider Attributes)
 - determining default message queue 121, 272, 273
 - object authority required 345
 - verifying attribute values 272
 - DSPSYSVAL (Display System Value) 275
 - verifying the current value 275
 - End Job Abnormal (ENDJOBABN) 104
 - ENDJOBABN (End Job Abnormal) 104
 - examples 259
 - Grant Object Authority (GRTOBJAUT)
 - example 253
 - no authority to run analysis 275
 - GRTOBJAUT (Grant Object Authority)
 - example 253
 - no authority to run analysis 275
 - HLDPTF (Hold Program Temporary Fix)
 - description 131
 - example 260
 - Hold Program Temporary Fix (HLDPTF)
 - description 131
 - example 260
 - object authority required 345
 - Order Supported Product PTFs (ORDSPTPTF)
 - description 119
 - example 262, 263
 - keeping service provider PTF database current 192
 - restrictions 121
 - ORDSPTPTF (Order Supported Product PTFs)
 - description 119
 - example 262, 263
 - keeping service provider PTF database current 192
 - restrictions 121
 - Print Error Log (PRTERLOG)
 - authorization to 275
 - PRTERLOG (Print Error Log)
 - authorization to 275
 - Release Program Temporary Fix (RLSPTF)
 - description 132
 - PTF distribution not successful 273
 - Remove Program Temporary Fix (RMVPTF) 121
 - Restore Licensed Program (RSTLICPGM) 112
 - RLSPTF (Release Program Temporary Fix)
 - description 132
- command, CL (*continued*)
 - RLSPTF (Release Program Temporary Fix) (*continued*)
 - PTF distribution not successful 273
 - RMVPTF (Remove Program Temporary Fix) 121
 - RSTLICPGM 112
 - Save Licensed Program (SAVLICPGM) 111
 - SAVLICPGM (Save Licensed Program) 111
 - Send Program Temporary Fix (SNDPTF)
 - description 133
 - sending PTFs to service requester 149
 - Send Program Temporary Fix Order (SNDPTFORD) 149
 - description 116
 - example 262
 - ordering cover letters 125
 - Send Service Request (SNDSRVRQS)
 - description 192
 - sending more than one problem 169
 - SNDLIC (Send License) 223
 - SNDPTF (Send Program Temporary Fix)
 - description 133
 - to service requester 149
 - SNDPTFORD (Send Program Temporary Fix Order)
 - description 116
 - example 262
 - ordering cover letters 125
 - SNDSRVRQS (Send Service Request)
 - description 192
 - sending more than one problem 169
 - Start Copy Screen (STRCPYSCN) 260
 - Start Subsystem (STRSBS) 274
 - STRCPYSCN (Start Copy Screen) 260
 - STRSBS (Start Subsystem) 274
 - Verify Communications (VFYCMN) 195
 - Verify Tape (VFYTAP) 195
 - VFYCMN (Verify Communications) 195
 - VFYTAP (Verify Tape) 195
 - Work with Active Jobs (WRKACTJOB) 260
 - Work with Alerts (WRKALR)
 - description 194
 - example 270
 - starting remote problem analysis 185
 - Work with Contact Information (WRKCNTINF)
 - changing cover letter language 125
 - example 271
 - Work with Distribution Queue (WRKDSTQ) 274
 - Work with License Information (WRKLCINF) 94, 103
- command, CL (*continued*)
 - Work with Problem (WRKPRB)
 - answering software problems 189
 - authorization to 275
 - changing problem status to run local analysis 186
 - description 157, 194
 - displaying problem details 176
 - example 259
 - examples 158
 - starting remote problem analysis 185, 187
 - Work with Program Temporary Fix (WRKPTF) 345
 - description 127
 - viewing PTF attributes 142
 - Work with Service Providers (WRKSRVPVD)
 - example 250, 252
 - Work with Service Requesters (WRKSRVRQS)
 - correcting machine information 275
 - determining default message queue 273
 - example 244
 - verifying attribute values 273
 - verifying information 272
 - Work with Supported Products (WRKSPTPRD)
 - adding product support 275
 - description 115
 - example 262
 - object authority required 345
 - verifying product support 273
 - WRKACTJOB (Work with Active Jobs) 260
 - WRKALR (Work with Alerts)
 - description 194
 - example 270
 - starting remote problem analysis 185
 - WRKCNTINF (Work with Contact Information)
 - changing cover letter language 125
 - example 271
 - WRKDSTQ (Work with Distribution Queue) 274
 - WRKLCINF (Work with License Information) 94, 103
 - WRKPRB (Work with Problem) 259
 - authorization to 275
 - description 157, 194
 - displaying problem details 176
 - examples 158
 - starting remote problem analysis 185
 - WRKPTF (Work with Program Temporary Fix) 345
 - description 127
 - viewing PTF attributes 142
 - WRKSPTPRD (Work with Supported Products) 275
 - example 262
 - object authority required 345
 - verifying product support 273

- command, CL (*continued*)
 - WRKSRVPVD (Work with Service Providers)
 - example 250
 - WRKSRVRQS (Work with Service Requesters)
 - correcting machine information 275
 - verifying attribute values 273
- commands
 - Create Product Definition
 - example 50
 - distribution 219
 - fast path 219
 - remote 229
 - using remote 229
- communications 8
 - configuring 237
 - setting up
 - service provider example 240, 241
 - service requester 248
 - service requester example 247
 - verifying 195
- compliance considerations for APIs 104
- compliance type
 - keyed 95, 104
 - operator action 95, 104
 - warning 95, 104
- concurrent usage licenses 105
- concurrent usage type 95
- conditions
 - change request 206
- conditions for using problem operations 164
- configuration
 - problems with SNA Distribution Services (SNADS) 274
 - verifying
 - secondary system name 243
 - verifying SNADS 237
- configuration lists
 - working with 18
- Configure Distribution Services (CFGDSTSRV) command
 - correcting SNADS problems 274
 - SNADS setup example 242, 249
 - using to set up SNADS 235
- configuring
 - communications and setup 237
 - distribution queues 235, 242
 - distribution services 235, 242, 249
 - routing table entries 236, 243
 - System Manager 16
 - verifying secondary system name 236
- contact information
 - verifying 14
 - working with 125
 - example 271
- control point name 8, 237
- controller description
 - creating
 - service provider example 241
 - service requester example 248
 - working with 12
- coordinating
 - problem management operations 153
- Copy Program Temporary Fix (CPYPTF) command
 - restrictions 125
- Copy PTF to Save File (CPYPTFSAVF) command
 - authorized IBM-supplied user profiles 343
 - description 123
 - receiving PTFs 124
- copy screen
 - starting 260
- copying
 - entries
 - service provider 28
 - service requester 20
 - program temporary fix 125
 - PTF save file 123
 - service requester entries on the service provider 21
- Copying
 - entries
 - service provider 29
 - corequisite PTF 144
- correcting problems
 - with System Manager 272
- cover letter
 - checking for special instructions 274
 - displaying 125
 - ordering 116, 117
 - printing 126
 - SNDPTFORD (Send Program Temporary Fix Order) 116
 - description 116
 - working with 125
- cover letters
 - preparing 144
- CPYPTF (Copy Program Temporary Fix) command
 - restrictions 125
- CPYPTFSAVF (Copy PTF to Save File) command
 - authorized IBM-supplied user profiles 343
 - description 123
 - object authority required 345
 - receiving PTFs 124
- Create Controller Description (APPC) (CRTCTLAPPC) command
 - example 241
- Create Line Description (Token-Ring) (CRTLINTRN) command
 - example 241, 248
- Create Message File (CRTMSGF) command
 - 49
- Create Product Definition (CRTPRDDFN) command
 - authorized IBM-supplied user profiles 343
- Create Product Load (CRTPRDLOD) command
 - authorized IBM-supplied user profiles 343
 - specifying folders 56
 - using multiple libraries 55
- Create Program Temporary Fix (CRTPTF) command
 - authorized IBM-supplied user profiles 343
 - object authority required 345
 - using different release levels 142
- Create Program Temporary Fix Package (CRTPTFPKG) command
 - authorized IBM-supplied user profiles 344
 - copying PTFs to tape 149
 - description 136
 - distributing PTFs 149
 - packaging PTFs 136
 - sending PTFs electronically 136
- Create User Profile (CRTUSRPRF) command
 - displays 252
 - example 252
 - for problem management tasks 29
 - creating 241
 - APAR 181
 - authorization list for service provider commands 17
 - authorized program analysis report (APAR) 181
 - change request description 198
 - controller description
 - example 241
 - service provider example 241
 - controller descriptions
 - service requester example 248
 - directory entries
 - on the service provider 236
 - on the service requester 236
 - service provider 250
 - service provider example 246
 - service requester 20, 244
 - service requester example 251
 - directory entry 28
 - distribution queue 235, 249
 - sending distribution feedback 236
 - service provider example 242
 - line description
 - example 241, 248
 - message file 49
 - message files and messages 78
 - problem records
 - Analyze Problem (ANZPRB) command 164
 - attributes not set correctly 272
 - from alerts at the service provider 11
 - setting up 153
 - system-detected problems 153
 - user-detected problems 153
 - when alerts received 182
 - when service requests received 182
 - product definition 49, 343
 - product load 52, 55, 56, 343
 - product options 50
 - program temporary fix (PTF) 142, 343
 - program temporary fix package 136, 149, 344
 - programs using output files 128

- creating (*continued*)
 - routing table entries
 - on the service provider 236, 243
 - on the service requester 236, 249
 - user profile 29, 252
 - user profile remote problem analysis 29, 252
 - CRTCTLAPPC (Create Controller Description APPC) command
 - example 241
 - CRTLINTRN (Create Line Description (Token-Ring Network)) command
 - example 241
 - CRTLINTRN (Create Line Description (Token-Ring)) command
 - example 248
 - CRTMSGF (Create Message File) command 49
 - CRTPRDDFN (Create Product Definition) command
 - authorized IBM-supplied user profiles 343
 - object authority required 345
 - CRTPRDLOD (Create Product Load) command
 - authorized IBM-supplied user profiles 343
 - object authority required 345
 - specifying folders 56
 - using multiple libraries 55
 - CRTPTF (Create Program Temporary Fix) command
 - authorized IBM-supplied user profiles 343
 - object authority required 345
 - using different release levels 142
 - CRTPTF (Create PTF) command
 - directories
 - development environment 146
 - CRTPTFPKG (Create Program Temporary Fix Package) command
 - authorized IBM-supplied user profiles 344
 - copying PTFs to tape 149
 - packaging PTFs 136
 - sending PTFs electronically 136
 - CRTUSRPRF (Create User Profile) command
 - example 252
 - for problem management tasks 29
 - cumulative PTF package 116
 - customizing
 - problem description 162
- D**
- data
 - type specific 208
 - database file
 - packaging 39, 46
 - defective PTF 120
 - defining features for license management 96
 - definition
 - corequisite relationship ix
 - creating 49
 - problem log 182
 - definition (*continued*)
 - product 49
 - Delete Problem (DLTPRB) command 162
 - Delete Program Temporary Fix (DLTPTF) command
 - description 132
 - save file management 125
 - deleting
 - a product using an exit program 76
 - directory entries 236
 - entries
 - service requester 20
 - problem records 162
 - APARs (authorized program analysis reports) 167
 - program temporary fix 125
 - PTFs 125
 - deleting APAR data
 - menu selection 167
 - delivering PTFs
 - electronically 118, 119
 - limitations 121
 - on tape 118, 119
 - service link 118, 119
 - dependencies on other PTFs 143
 - details 211
 - detecting problems 180
 - developing
 - PTFs 146
 - developing PTF exit programs 147
 - development folders 43
 - development libraries 43
 - development process
 - for complex product 40
 - directories
 - assigning names
 - restrictions 63
 - naming conventions 61
 - product
 - packaging 61
 - product directories 61
 - product home directories 61
 - product load
 - specifying directory list 63
 - PTFs
 - names 146
 - directory entry
 - adding
 - service provider example 246
 - service requester example 251
 - creating
 - automatically for distribution feedback 237
 - directory 237
 - service provider example 246
 - service requester example 251
 - creating automatically 28
 - creating on the service provider 239
 - creating on the service requester 238
 - distribution feedback 237
 - requirements 237
 - disk space
 - cleaning up 12
 - minimizing 6
 - reclaiming 12, 164
 - requirements 6
 - storage management 6
 - display activity exit program 311
 - display APAR data
 - menu selection 167
 - Display Job Log (DSPJOBLOG) command
 - applying PTF 274
 - checking job log 274
 - displaying results of service request 266
 - Display Messages (DSPMSG) command
 - displaying the service provider message queue 185
 - example 270
 - Display Network Attributes (DSPNETA) command
 - using to verify system name 10
 - Display Problem (DSPPRB) command
 - displaying a list of problems 176
 - Display Problems (DSPPRB) command
 - description 159
 - Display Program Temporary Fix (DSPPTF) command
 - checking PTF status 124, 273
 - description 128
 - example 129, 261
 - verifying PTF status 274
 - Display PTFs and IBM Query 129
 - Display Service Provider Attributes (DSPSRVPVDA) command
 - description 19
 - determining default message queue 121, 273
 - object authority required 345
 - verifying attribute values 272, 273
 - Display System Value (DSPSYSVAL) command 21
 - verifying the current value 275
 - displaying
 - activity details 205
 - alerts 12
 - change request description activity details 205
 - conditions 206
 - cover letters 125
 - details
 - PTFs 148
 - entries
 - service requester 20
 - sphere of control 12
 - history 176
 - job log 266
 - network attributes 10
 - nodes 206
 - output 230
 - problem analysis results 155
 - problem records
 - APAR (authorized program analysis report) 167
 - problems 159, 176
 - product support information 25
 - PTF
 - information 190, 261
 - status 128
 - service provider
 - entries 27
 - service provider attributes 19, 345
 - spooled files 156
 - system value 275

- displaying messages
 - example 270
 - displays
 - Add Service Provider 250
 - Add Service Requester 20, 244
 - Change Network Attributes (CHGNETA) 240, 241
 - Change Problem (CHGPRB) 161
 - change request description 199
 - Change Service Provider Attributes (CHGSRVPVDA) 18
 - CHGNETA (Change Network Attributes) 240
 - CHGPRB (Change Problem) 161
 - CHGSRVPVDA (Change Service Provider Attributes) 18
 - Create User Profile (CRTUSRPRF) 252
 - CRTUSRPRF (Create User Profile) 252
 - Delete Problem (DLTPRB) 162
 - Display Directory Entry Details 246, 252
 - Display Problem Details 153
 - Display Problem History 176
 - Display Problems (DSPPRB) 159
 - Display Program Temporary Fix (DSPPTF) 129
 - DSPPTF (Display Program Temporary Fix) 129
 - Order Supported Product PTFs (ORDSPTPTF) 120
 - ORDSPTPTF (Order Supported Product PTFs) 120
 - Select and Sequence Fields 130
 - Send PTF Order (SNDPTFORD) 118
 - Send Service Request (SNDSRVRQS) 193
 - SNDPTFORD (Send PTF Order) 118
 - SNDSRVRQS (Send Service Request) 193
 - Work with Problem (WRKPRB) 157
 - Work with PTF Information 190
 - Work with Service Providers 27
 - Work with Service Requesters 19
 - Work with Supported Products 23
 - WRKPRB (Work with Problem) 157
 - distributing
 - products 113
 - PTFs 149
 - directory entry for distribution feedback 237
 - electronically 133
 - methods 22
 - not being sent 273
 - service requester 274
 - to multiple service requesters 133
 - using object distribution 122, 235
 - using SNA distribution services (SNADS) 235
 - distributing PTFs
 - with Management Central 149
 - distribution
 - commands 219
 - directory entries for feedback 237
 - fast path commands 219
 - distribution (*continued*)
 - feedback from the service requester 252
 - product 111
 - queue
 - adding 235, 242
 - configuring 242
 - creating 235, 242
 - requirements 237
 - working with 274
 - services
 - configuring 235
 - distribution services
 - configuring 242, 249
 - DLTPRB (Delete Problem) command 162
 - DLTPRDDFN (Delete Product Definition) command
 - authorized IBM-supplied user profiles 344
 - object authority required 345
 - DLTPRDLOD (Delete Product Load) authorized IBM-supplied user profiles 344
 - DLTPRDLOD (Delete Product Load) command
 - object authority required 345
 - DLTPTF (Delete Program Temporary Fix) command
 - description 132
 - save file management 125
 - downloading PTF
 - from IBM service support (*IBMSRV) 121
 - DSPJOBLOG (Display Job Log) command
 - applying PTF 274
 - checking job log 274
 - displaying results of service request 266
 - DSPMSG (Display Messages) command
 - displaying the service provider message queue 185
 - example 270
 - DSPNETA (Display Network Attributes) command
 - using to verify system name 10
 - DSPPRB (Display Problem) command
 - displaying a list of problems 176
 - DSPPRB (Display Problems) command
 - description 159
 - DSPPTF (Display Program Temporary Fix) command 124
 - checking PTF status 273
 - description 128
 - example 129, 261
 - IBM Query 128
 - verifying PTF status 274
 - DSPPTF command 148
 - DSPSRVPVDA (Display Service Provider Attributes) command 19
 - determining default message queue 121, 272
 - IBM Query 121
 - object authority required 345
 - verifying attribute values 272, 273
 - DSPSYSVAL (Display System Value) command
 - verifying the current value 275
 - dynamic naming 41
- ## E
- electronic customer support service link
 - delivering PTFs 118
 - receiving PTFs 25
 - End Job Abnormal (ENDJOBABN) command 104
 - end node
 - service requester
 - example 248
 - setting up alerts 11
 - ending
 - job abnormal 104
 - QSNADS subsystem 28
 - ENDJOBABN (End Job Abnormal) command 104
 - entitling
 - more than one service provider 27
 - systems to service 19
 - entries
 - adding 12
 - changing
 - ALRCTL (alert controller description) attribute 12
 - service requester
 - adding 20, 244
 - changing 20
 - copying 20, 21
 - creating 244
 - deleting 20
 - displaying 20
 - sphere of control 12
 - changing 12
 - displaying 12
 - removing 12
 - error log
 - information 180
 - printing 29, 275
 - error messages 74
 - example 129, 263
 - adding
 - directory entries on service provider 246
 - distribution queues on the service provider 242
 - distribution queues on the service requester 249
 - product support 262
 - routing table entries 243
 - routing table entries on the service requester 249
 - alert controller description (ALRCTL) attribute 33, 35
 - alert focal point 31
 - alert network attributes for a service provider 35
 - alert reported a system-detected problem 257
 - alert reported printer problem 269
 - alert support
 - for service providers 32
 - for service requester 33

example (*continued*)

- Change Network Attributes (CHGNETA) command 240
- Change Problem (CHGPRB) command 162
- CHGNETA (Change Network Attributes) command 240
- configuring network 237
- configuring secondary system name table 243, 250
- connecting to IBM service support (*IBMSRV) 34
- creating
 - controller description 241
 - controller descriptions on the service provider 241
 - controller descriptions on the service requester 248
 - directory entries on service provider 246
 - distribution queues on the service provider 242
 - line descriptions on a service provider 241
 - line descriptions on a service requester 248
 - product definition 49
 - product load 54
 - routing table entries 243
 - user profile for remote problem analysis 252
- creating line description 241
- Delete Problem (DLTPRB) command 163
- detecting and correcting a hardware problem 264
- directory entries
 - for a service requester 251
 - for distribution feedback 252
- display problems
 - output file support 159
- displaying messages 270
- end node 33
- exit program 277
- hardware service request 263
- how to finish packaging 82
- installing System Manager 16
- iSeries network 31
- line description
 - creating 241, 248
- network 31
- network attributes 33
 - for the service provider 241
 - for the service requester 248
- network ID 9
- network node server (NETSERVER) attribute 33
- packaging a product with directories 63
- packaging a product with folders 57
- problem reporting 259
- problem situations 257
- PTF information from IBM Query 131
- PTF order and distribution 262
- reporting
 - a hardware problem 263

example (*continued*)

- reporting (*continued*)
 - a problem 259
- send PTF order from service requester 271
- sending PTF order from service requester 270
- service provider attributes 33
- service provider network node server 248
- service requester end node 248
- setting up
 - a service requester 250
 - another service requester 253
 - communications on the service provider 240, 247
 - configuration 237
 - configuration example 237
 - list of service providers 250
 - network communications 237
 - SNADS on a service provider 242
 - SNADS on a service requester 249
- setting up a network
 - filled in work sheet 254
- setting up System Manager 32, 34
- system-detected problem
 - service request reported 259
- System/370 and iSeries network 34
- System/370 or System/390 and NetView 34
- user-detected performance problem 260, 267
- Work with Problem (WRKPRB) command 158, 159
- Work with Sphere of Control (WRKSOC) command 35
- working with alerts 270

exceeding the PTF order limit 121

exit program

- add activity 308
- change activity 310
- data 205
- description 72
- developing PTF 147
- display activity 311
- example 277
- functions for
 - checking a product 76
 - deleting a product 76
 - restoring a product 75
 - saving a product 75
- how to start 205
- postoperation 73
- preoperation 73
- product 72
- product management 74
- rules and restrictions 76
- run activity 314

F

- fast path
 - commands 219
- features
 - defining for license management 96

- file, database
 - packaging 39, 46
- filtering
 - alert 13
 - problems 176
- focal point
 - See also* alerts
 - alert filtering 13
 - problem filtering 165
- folders
 - development 43
 - missing
 - verification 76
 - PTF documents 146
- functions
 - exit program
 - checking a product 76
 - deleting a product 76
 - restoring a product 75
 - saving a product 75

G

- General Purpose Library (QGPL) 142
- global names 203
- Grant Object Authority (GRTOBJAUT) command
 - example 253
 - no authority to run analysis 275
 - setting up command authority 17
- granting authority
 - new commands 17
 - to use commands 29
 - object authority 17, 253, 275
- group assigned parameter
 - setting 162
- grouping
 - alerts 13
 - problems 176
- GRTOBJAUT (Grant Object Authority) command
 - example 253
 - no authority to run analysis 275
 - setting up command authority 17

H

- HIPER (high-impact pervasive) PTFs 120
- history
 - problem 175
- HLDPTF (Hold Program Temporary Fix) command
 - authorized IBM-supplied user profiles 344
 - description 131, 260
 - example 260
 - object authority required 345
- HLDPTF (hold PTF received) attribute
 - automatically holding PTFs received 131
 - definition 19
- Hold Program Temporary Fix (HLDPTF) command
 - description 131

- Hold Program Temporary Fix (HLDPTF)
 - command (*continued*)
 - example 260
 - object authority required 345
- hold PTF received (HLDPTF) attribute
 - automatically holding PTFs
 - received 131
- hold PTFs received (HLDPTF) attribute
 - definition 19
- holding
 - program temporary fix (PTF) 131, 190, 260

I

- IBM marketing support
 - QTILINE 8
- IBM Query licensed program 128
- IBM service support (*IBMSRV) 8
 - changing the route 29, 251
 - QESLINE 8
 - receiving PTFs 122
 - reporting problems 165
 - sending PTF orders 119
 - sending test requests 30
 - service providers entry 28
- IBM-supplied user profile
 - restricted commands 343
- identifier
 - PTF (program temporary fix) 142
- including objects in the PTF 143
- information
 - adding to PTF database 22
 - product 142
- installable object
 - definition 197
- installing
 - more than one release 22
 - multiple releases of a product 42
 - new release 22
 - PTF 151
 - service provider
 - System Manager 16
- installing Managed System Services
 - service provider 14
- iSeries example network 31

J

- job abnormal
 - ending 104
- job log
 - checking 274
 - displaying 266

K

- key
 - creating a license 102

L

- language
 - translating 41

- language feature support
 - adding 24
 - service provider 24
- language load
 - definition 53
 - description 52
 - using 41
- library
 - development 43
 - General Purpose Library (QGPL) 142
 - PTF 143
- library list
 - naming conventions 112
 - restoring 112
- license
 - releasing the use 103
- license information
 - adding 98
 - working with 94, 103
- license key
 - creating 102
- license management
 - application program interface (API)
 - calls 102
 - authority considerations for APIs 105
 - compliance considerations for APIs 104
 - overview 93
 - packaging features 96
 - packaging user-based priced products 96
 - planning 94
 - release-to-release considerations 96
 - requesting the use of a license 103
 - usage types
 - concurrent 95
 - registered 95
- license term strategy 94
- licensed program 128
 - saving 111
- licensed program offerings
 - adding support 23
- licenses
 - concurrent usage 105
 - registered usage 107
- load
 - code 52
 - language 52
 - product 52
- local location name 8
- local problem analysis required 185
- location list
 - remote 18
- location name 8
- location password 18
- log
 - problem, working with 182
- log remote problems (LOGRMTPRB)
 - attribute
 - definition 18
 - example 257, 269
 - setting 18, 153
 - using 18
- logging
 - problems 153

- LOGRMTPRB (log remote problems)
 - attribute
 - definition 18
 - example 257, 269
 - setting 18, 153
 - using 18

M

- machine information
 - serial number 21
 - type 21
- maintenance
 - complex product 39
 - preventive 116
 - problem log 163
 - restoring disk space 164
- Managed System Services
 - starting 26
- management
 - installation 111
 - product 111
- Management Central
 - distributing PTFs 149
- MANAGESYS (Manage Your System, Users, and Devices) menu
 - Operational Assistant menu
 - automatic cleanup functions 6
 - reclaiming disk space from deleted alerts 12
 - reclaiming storage from deleted problems 163
 - reclaiming storage from deleted PTFs 132
- managing
 - PTFs 124, 125
- map for matching setup parameter values 239
- mapping
 - setup commands 6
- marketing support line QTILINE 8
- marking objects for packaging 78
- matching map of parameter values 239
- MAXDST (maximum distribution)
 - attribute
 - definition 19
 - example 270
- maximum distribution (MAXDST)
 - attribute
 - definition 19
 - example 270
- message
 - cannot find on service provider 273
 - change requests, submitted 215
 - CPF3D95 86
 - CPF3D98 86
 - CPF9E18 104
 - CPH9E17 104
 - creating 78
 - displaying, example 270
 - distribution error 237
 - distribution feedback 237
 - error 74
 - error on test request 31
 - service provider queue 258
 - service requester status 30
 - submitted change requests 215

- message file
 - creating 49, 78
- message queue (MSGQ) attribute
 - definition 19
 - examples 257, 269
 - setting 18
- minimizing disk space 6
- missing
 - folders 76
 - objects 76
- mode
 - APPN communications
 - #INTER 8
- MSGQ (message queue) attribute
 - definition 19
 - examples 257, 269
 - setting 18
- multiple releases of a product
 - installing 42
 - supporting 42

N

- names
 - global 203
 - standard 203
 - system
 - LCLCPNAME (local control name) 10
 - LCLLOCNAME (local location name) 10
 - Local control point name (LCLCPNAME) 10
 - Local location name (LCLLOCNAME) 10
 - SYSNAME (system name) 10
 - system name (SYSNAME) 10
 - verifying system 10
- naming
 - conventions
 - library list 112
 - restoring 112
 - dynamic 41
 - network systems 8
 - PTF
 - directories 146
- NETSERVER (network node server)
 - attribute
 - changing
 - example 33
- NETSERVER (network server) attribute
 - changing 11
 - example 33
- network
 - Display Network Attributes (DSPNETA) command 10
 - DSPNETA (Display Network Attributes) command 10
 - examples 31
 - naming systems 8
 - nodes 11, 248
 - problem tracking 175
 - setting up for System Manager 1

- network attributes
 - See also* attribute
 - alert controller description (ALRCTLD)
 - ALRCTLD (alert controller description) attribute 12
 - alert status *UNATTEND 11
 - ALRCTLD (alert controller description) 12
 - changing 11, 240
 - setting up alerts 11
 - displaying 10
 - example 33
- network ID
 - characteristics 9
 - definition 9
 - naming systems 8
 - registering 10
 - SNA distribution services (SNADS) 237
 - SNADS (SNA distribution services) 237
- network server (NETSERVER)
 - attribute 33
- node
 - end 11, 248
 - network 11, 248
- node list 202
- node names 202
- nodes
 - displaying 206
 - working with 213

O

- object
 - authority required for objects used by commands 345
 - including in PTF 143
 - missing
 - verification 76
 - saving 25
- object authority
 - common object commands 345
 - granting 17, 253
 - for commands 275
- object distribution
 - definition 235
 - setting up 235
- operating system release requirements 5
- Operational Assistant
 - automatic cleanup functions
 - disk space management 6
 - reclaiming disk space from deleted alerts 12
 - reclaiming disk space from deleted problems 163
 - reclaiming disk space from deleted PTFs 132
 - creating a problem record 181
 - definition 181
 - saving information to report a problem 181
 - USERHELP menu 181
- operations
 - controlling 197
- OPTION parameter 50

- options
 - adding details 52
 - product
 - *BASE 50
 - creating 50
 - defining 50
 - used to define features 96
- Order Supported Product PTFs (ORDSPPTF) command
 - description 119
 - example 262, 263
 - keeping service provider PTF database
 - current 192
 - object authority required 345
 - restrictions 121
- ordering
 - cover letters 116, 117
 - process changes for user-based pricing 94
 - PTFs
 - commands 116
 - for supported products 263
 - from IBM service support (*IBMSRV) 119
 - limitations 121
 - overview 116
 - supported product PTFs 119, 121
- ORDSPPTF (Order Supported Product PTFs) command
 - description 119
 - example 262, 263
 - keeping service provider PTF database
 - current 192
 - object authority required 345
 - restrictions 121
- output
 - displaying 230
- output file
 - as program data 159
- output files
 - using problem log 159

P

- Package Product Option (PKGPRDOPT)
 - command 80, 82
 - authorized IBM-supplied user profiles 344
- packaging
 - a follow-on release 46
 - applications with more than one library
 - example 55
 - complex product considerations 37
 - database files 39, 46
 - example
 - product with folders 57
 - features for license management 96
 - follow-on release 46
 - marking objects 78
 - product 80
 - concepts 37
 - option 82
 - tips and techniques 86
 - with folders or documents 56
 - with more than one library 55

- packaging (*continued*)
 - product with more than one library
 - example 55
 - products
 - with directories 61
 - user-based priced products for license management 96
- packaging features for license management 96
- packing
 - product load
 - assigning directory names 63
- parameter descriptions 233
- password
 - for remote problem analysis not correct 275
 - location 18
- PC code 57
- performing preventive maintenance on the OS/400 system 116
- PKGPRDOPT (Package Product Option)
 - command 80, 82
 - authorized IBM-supplied user profiles 344
 - object authority required 345
- planning
 - storage space 6
 - system support 4
- postoperation exit program 73
- preoperation exit program 73
- prepackaging considerations for complex products 37
- preparing
 - mapping setup commands 6
 - service providers for setup 1
 - service requesters 1
 - setting up your systems 1
- prerequisite PTF 143
- prerequisite relationship 143
- preventive maintenance 116
- previewing PTF 117
- Print Error Log (PRTERRLOG) command
 - authorization to 29, 275
- printing
 - cover letters 126
 - error log 29, 275
 - product support information 25
- problem
 - analyzing 11, 150
 - definition 179
 - example from message queue 260
 - example performance problem 261, 266
 - analyzing local
 - from the problem log 164
 - analyzing remote
 - from command line 186
 - from message queue 185
 - from problem log 185, 187
 - from the problem log 164
 - from Work with Alerts
 - display 185, 186
 - introduction 179
 - preparing for 184
 - system-detected problems 184
 - user-detected problems 186
- problem (*continued*)
 - answering
 - hardware 165, 192
 - introduction 188
 - software 165, 189
 - cannot run problem analysis 275
 - changing 161
 - changing information 161
 - detected by the system
 - reporting 182
 - detecting 180
 - displaying 159, 176
 - displaying details 176
 - equipment
 - reporting 192
 - filtering 176
 - grouping 176
 - history 175
 - local problem analysis required 185
 - receiving PTFs as answers 122
 - reporting 165
 - restrictions 193
 - to the service provider 151
 - system-detected 153, 180
 - table of status changes and operations 174, 175
 - tracking 175
 - types of 180
 - user-detected 153, 181
 - with setup verification 31
 - working with 157, 165, 259
 - analyze problem option 164
 - answer problem option 165
 - close problem option 166
 - created on service requester 151
 - delete APAR data option 167
 - display APAR data option 167
 - example 257
 - example performance problem 266
 - filtering 176
 - grouping 176
 - options 164
 - recovery procedures option 165
 - save APAR data option 166
 - save APAR library option 167
 - send response option 166
 - verification procedures option 165
- problem analysis
 - cannot run remote 275
 - commands 194
 - completing 187
 - creating
 - user profile for remote analysis 29, 252
 - definition 179
 - filtering 176
 - for an installed product 113
 - granting authority
 - commands running remote analysis 30
 - grouping 176
 - password for remote not correct 275
 - remote failure 275
 - results
 - answer list 188
- problem analysis (*continued*)
 - results (*continued*)
 - displaying 155
 - isolation list 188
 - partial isolation list 188
 - point of failure list 187, 188
 - possible causes list 183, 187
 - recovery list 188
 - symptom string 183, 187
 - verification list 188
 - running
 - from the problem log 164
 - OPENED problem 168
 - remote and unattended 11, 183
 - system-detected problem 164
 - user-detected problem 164
 - running remote
 - introduction 179
 - preparing for 184
 - unattended 179
 - setting up authority for 29
 - starting remote
 - from command line 186
 - from message queue 185
 - from problem log 185, 187
 - from Work with Alerts
 - display 185, 186
 - system-detected problems 184
 - user-detected problems 186
 - user profile not properly authorized 275
 - problem description
 - changing 162
 - problem hold interval (QPRBHLDITV)
 - system value 163
 - problem log
 - answering problems 165, 188
 - changing text fields 161
 - cleaning up disk space 12
 - closing problem record 166
 - deleting APAR data 167
 - displaying APAR data 167
 - filtering 176
 - grouping 176
 - maintenance 163
 - output files 159
 - reporting problems 165, 192
 - saving
 - APAR data 166
 - APAR library 167
 - searching 145
 - setting problem severity 161
 - starting remote problem analysis 185
 - status changes on service requester 175
 - verification procedure 165
 - working with 153
 - working with definition 182
 - problem management
 - APIs 195
 - coordinating operations 153
 - working with problem 164
 - filtering 176
 - grouping 176
 - working with problems
 - command description 157

- problem notifications
 - alerts
 - creating problem record 153
 - definition 183
 - system-detected problems 10, 183
 - receiving 181
 - running remote
 - unattended 183
 - service requests
 - creating problem record 153
 - definition 183
- problem records
 - adding
 - answer information 188
 - problem analysis results 168
 - ANSWERED status 171
 - answering
 - results 166
 - sending PTFs 171
 - associated spooled files 156
 - change request descriptions 157
 - changing
 - status 167
 - text 161
 - closing 166
 - creating
 - Analyze Problem (ANZPRB)
 - command 164
 - APARs 181
 - attributes not set correctly 272
 - from alerts at the service
 - provider 11
 - setting up 153
 - system-detected problem 153
 - user-detected problem 153
 - when alerts received 182
 - when service requests
 - received 182
 - deleting 162
 - deleting APAR data 167
 - displaying APAR data 167
 - information in 153, 180
 - not created when alerts received 272
 - operations that change status 173
 - performing operations 155, 157
 - reporting
 - status 192
 - results of PTF search 165
 - running recovery procedures 165
 - saving
 - APAR data 166
 - APAR library 167
 - sending
 - response 166, 172
 - setting problem severity 161
 - status
 - when reporting 165
 - when responding 166, 172
 - status changes
 - ANSWERED status 171
 - CLOSED status 173
 - OPENED status 167
 - PREPARED status 169, 186
 - READY status 168
 - SENT status 170
 - VERIFIED status 172
 - verifying problem corrected 165
- problem records (*continued*)
 - working with 164
- problem severity values
 - setting 161
- problem status
 - ANSWERED 171
 - CLOSED 173
 - OPENED 167
 - PREPARED 169
 - READY 168
 - SENT 170
 - VERIFIED 172
- product
 - application environment for
 - complex 39
 - changing product object descriptions
 - example 78
 - creating a product load
 - example 54
 - development process for complex 40
 - distributing 111, 113
 - example 50, 54
 - how to finish packaging 82
 - packaging a product with
 - directories 63
 - packaging a product with
 - folders 57
 - example of a user-based priced
 - product 96
 - exit program 72
 - information 142
 - installation 111
 - load
 - creating 55, 56, 343
 - maintenance 39
 - management 111
 - message file
 - creating 49
 - option
 - checking 76
 - packaging 82
 - options
 - *BASE 50
 - creating 50
 - defining 50
 - packaging 80
 - concepts 37
 - containing document objects 56
 - directories 61
 - prepackaging considerations 37
 - restrictions 48
 - using folders 56
 - with more than one library 55
 - problem analysis 113
 - product definition
 - creating 49
 - example 50
 - product load 80
 - release level (RLS) identifiers 50
 - servicing 141
 - testing 112
 - typical product 50
 - user-specified folder names 57
 - user-specified library names 57
- Product (*continued*)
 - recovery 113
 - product availability object 25
 - saving backup copy 25
 - product definition
 - creating 343
 - product distribution 111
 - product exit program interface 74
 - product exit programs 72
 - product information 142
 - product installation 111
 - product load
 - creating 55, 56, 343
 - packaging
 - assigning directory names 63
 - product management 111
 - exit program 74
 - product management exit program 74
 - product message file
 - creating 49
 - product option
 - checking 76
 - packaging 82
 - product owner
 - service information 141
 - product packaging
 - restrictions 48, 81
 - product structures
 - Check Product Option (CHKPRDOPT)
 - command 86
 - checking for structure problems 86
 - product support
 - adding 262, 275
 - installing new release 22
 - removing information 25
 - setting up service provider 22
 - supporting more than one release 22
 - system performance when adding 23
 - working with 22
 - working with example 262
 - profile
 - IBM-supplied
 - restricted commands 343
 - program
 - processing authority 349
 - program temporary fix 142
 - applying 274
 - copying 125
 - creating 142, 343
 - deleting 125
 - holding 131, 260
 - releasing 132, 273
 - removing 121
 - sending 149
 - working with 142, 345
 - program temporary fix order
 - sending 125
 - program temporary fix package
 - creating 136, 149, 344
 - programming
 - command reference 233
 - programming requests for price
 - quotations (PRPQS)
 - adding support 23
 - PRPQ
 - adding support 23

PRTERLOG (Print Error Log) command
 authorization to 29, 275
 PTF
 adding
 information to problem record 190
 Apply Program Temporary Fix (APYPTF) command 274
 APYPTF (Apply Program Temporary Fix) command 274
 backup and recovery 149
 checking requisites 122
 collecting information 129
 command restrictions 121, 124
 corequisite 144
 corequisites 122
 cover letters
 checking for special instructions 274
 displaying 125
 preparing 144
 printing 126
 Create Program Temporary Fix Package (CRTPTFPKG)
 command 136
 creating 142
 CRTPTFPKG (Create Program Temporary Fix Package)
 command 136
 cumulative package 116
 database
 adding information 25
 setting up 22, 275
 defective notifications 120
 deleting 125
 delivering
 electronically 118, 119
 limitations 121
 on tape 118, 119
 service link 118, 119
 dependencies 143
 developing exit programs 147
 directories
 development environment 146
 names 146
 stream files 146
 displaying 128
 detailed information 261
 information 190
 distributing 273
 electronically 133
 methods 22
 to multiple service requesters 133
 using object distribution 122
 when PTFs do not arrive at service requester 274
 Work with PTF (WRKPTF)
 command 149
 WRKPTF (Work with PTF)
 command 149
 distribution requisites 122
 document in folder 146
 documents in folders 146
 downloading from IBM service support (*IBMSRV) 121
 exceeding the order limit 121
 fast path commands 219

PTF (continued)
 functions, using 115
 high-impact pervasive (HIPER)
 notifications 120
 HIPER (high-impact pervasive)
 notifications 120
 holding 190
 identifier 142
 including objects 143
 information
 clean up 152
 information to PTF database 22
 installing 151
 library 143
 managing 125
 ordering
 commands 116
 definition 116
 for supported products 263
 from IBM service support (*IBMSRV) 119
 limitations 121
 overview 116
 overview 115
 package 116
 prerequisite 121, 143
 prerequisites 122
 previewing 117
 problem
 not solved by applying 274
 rebuilding 149
 receiving
 from IBM service support (*IBMSRV) 122
 from OS/400 service providers 122
 from tape 123
 overview 121
 QSNADS subsystem 28, 274
 restrictions 124
 releasing 190
 removing information from problem record 190
 requisites 124
 restrictions 121
 save file
 copying 123
 save file names 123
 save file processing 275
 saving and restoring 152
 searching
 problem analysis results 188
 searching for
 answering problems 165
 matching problem symptoms 189
 problem in ANSWERED status 172
 sending
 Create Program Temporary Fix Package (CRTPTFPKG) 136
 Create Program Temporary Fix Package (CRTPTFPKG)
 command 136
 do not arrive at service requester 274
 electronically 136

PTF (continued)
 sending (continued)
 orders to IBM service support (*IBMSRV) 119
 to multiple service requesters 133
 special packages 121
 cumulative PTF package 116
 hardware service planning package 116
 preventive service planning package 116
 PTF summary package 116
 status
 description 191
 displaying 128
 of PTF 128
 when answering problems 166
 storing 121
 superseded 143, 144
 symptom strings 145
 testing 148
 transferring from tape 121
 working with
 cover letters 125
 information 165
 save files 124, 125

Q

QAUTOVRT (automatic virtual device configuration) system value
 changing 30
 definition 29
 example 253
 not set correctly 275
 QESLINE
 *IBMSRV (IBM service support) 8
 IBM service support (*IBMSRV) 8
 QGPL (general purpose library) 23
 QLZAREQ (request license) API 102, 103
 QLZAREQ (Request License) API 103
 QLZARLS (release license) API 102, 103
 QLZARLS (Release License) API 103
 QLZARTV (retrieve license information) API 102
 QLZARTV (Retrieve License Information) API 103
 QNSADDCM (Add Change Management Activity) API
 description 292
 QPRBHLDTIV (problem hold interval) system value 163
 QRMTSIGN (remote sign-on verification) system value
 changing 30, 253
 definition 29
 example 253
 not set correctly 275
 QSRLNBR (system serial number) system value 21
 QSRVBAS (basic security representative) user profile
 receiving files 123
 QSRVBAS (basic service representative) user profile 236

QSVSM
 user profile authorities 369

QTLINE, IBM marketing support 8

Query licensed program 128

queue
 distribution
 adding 235
 configuring 235
 creating 235
 requirements 237
 sending distribution feedback 236, 237
 service provider example 242
 service requester example 249
 SNA Distribution Services (SNADS) problems 274
 working with 274

message
 basic service representative (QSRVBAS) 274
 definition on service provider 19
 QSRVBAS (basic service representative) 274
 QSYSOPR (system operator) 274
 starting problem analysis 185
 system operator (QSYSOPR) 274
 service requester example 249

Queue
 message
 service provider example 258

R

rebuilding PTF 149

receiving
 problem notification 181

PTF
 from IBM service support (*IBMSRV) 122
 from OS/400 service provider 122
 from tape 123
 overview 121
 QSNADS subsystem 123
 restrictions 124

reclaiming disk space
 from deleted alerts 12
 from deleted problem records 164
 from deleted PTF save files and information 132

recovery procedure
 running 165

reference
 command 233

registered usage license 107

registered usage type 95

registering
 network ID 10

release
 packaging a follow-on 46

release level
 considerations 22
 planning support 4

Release License (QLZARLS) API 103

Release Program Temporary Fix (RLSPTF)
 command
 authorized IBM-supplied user profiles 344

Release Program Temporary Fix (RLSPTF)
 command (*continued*)
 description 132
 object authority required 345
 PTF distribution not successful 273

releases
 multiple 42

releasing
 program temporary fix 132, 273
 program temporary fix (PTF) 190

releasing the use of a license 103

remote commands, using 229

remote location list 18

remote problems
 local problem analysis required 185

remote sign-on verification (QRMTSIGN)
 system value
 changing 30, 253
 definition 29
 example 253
 not set correctly 275

Remove Program Temporary Fix (RMVPTF) command 121

removing
 entries
 sphere of control 12
 product support information 25
 program temporary fix 121
 PTF information
 from problem record 190

repackaging considerations 46

report capabilities 210

reporting
 problem
 from the service provider 192
 to a service provider 151, 165
 problems detected by the system
 manually 182
 restrictions 193
 to IBM service support (*IBMSRV) 165

Request License (QLZAREQ) API 103

requesting the use of a license 103

requirements
 Advanced Peer-to-Peer Networking (APPN) 7
 disk space 6
 system at different release level 4
 System Manager 7

requisites for PTF 122

Restore APAR Data (RSTAPARDTA)
 command
 authorized IBM-supplied user profiles 344

Restore Licensed Program (RSTLICPGM)
 command 112

restoring
 a product using an exit program 75
 library list 112
 naming conventions 112
 PTF 152

restrictions
 exit program 76
 product packaging 48

results
 displaying 155

results (*continued*)
 problem analysis
 answer list 188
 isolation list 188
 partial isolation list 188
 point of failure list 188
 possible causes list 183, 187
 recovery list 188
 symptom string 183, 187
 verification list 188

Retrieve License Information (QLZARTV)
 API 103

RLSPTF (Release Program Temporary Fix)
 command
 description 132
 object authority required 345
 PTF distribution not successful 273

RLSPTF (Release PTF) command
 authorized IBM-supplied user profiles 344

RMVPTF (Remove Program Temporary Fix) command 121

root folder document 57

routing errors for SNA Distribution Services (SNADS) 274

routing table entry
 adding 236
 configuring 243
 creating
 on the service provider 236, 243
 on the service requester 236, 249
 for sending distribution feedback 236

requirements 237

service provider example 243

service requester example 249

RSTAPARDTA (Restore APAR Data)
 command
 authorized IBM-supplied user profiles 344

rules
 exit program 76

run activity exit program 314

running
 change request 208
 local analysis for problem in SENT status 186
 problem analysis
 from the problem log 164
 OPENED problem 168
 remote and unattended 11, 183
 system-detected problem 164
 user-detected problem 164

recovery procedures 165

service requester
 local problem analysis 186

S

save and restore operations
 restore library (RSTLIB)
 operation 124
 RSTLIB (restore library)
 operation 124
 save library (SAVLIB) operation 124
 SAVLIB (save library) operation 124

- save APAR data
 - authorized program analysis report (APAR)
 - definition 166
 - menu selection 166, 167
- save files
 - working with 124
- Save Licensed Program (SAVLICPGM)
 - command 111
- Save Object (SAVOBJ) command
 - description 25
- saving
 - a product using an exit program 75
 - licensed program 111
 - object 25
 - problem log 166, 167
 - problem records
 - APAR (authorized program analysis report) library 167
 - data 167
 - product availability object 25
 - PTF 152
- SAVLICPGM (Save Licensed Program)
 - command 111
- SAVOBJ (Save Object) command
 - description 25
- scheduling
 - activities 201
- searching
 - problem log 145
- searching for PTFs
 - to answer problem 165, 188
- secondary system name
 - configuring 236, 243
- secondary system name table
 - requirements 237
- security 29
 - Grant Object Authority (GRTOBJAUT)
 - command
 - setting up command authority 17
 - service provider
 - introduction 17
 - location password 18
 - remote location list 18
 - service requester is not entitled to service 275
- select and sequence field
 - AS/400 Query display 130
- Send Program Temporary Fix (SNDPTF)
 - command
 - authorized IBM-supplied user profiles 344
 - description 133
 - object authority required 345
 - sending PTFs to service requester 149
- Send PTF Order (SNDPTFORD)
 - command 149
 - description 116
 - example 262
 - ordering cover letters 125
- send PTFs automatically (SNDPTFAUTO)
 - attribute
 - controlling PTF distribution 123
 - definition 19
 - example 259, 270
 - not set correctly 273
 - send PTFs automatically (SNDPTFAUTO)
 - attribute (*continued*)
 - releasing PTFs 132
 - setting 18
 - setting for manual intervention 22
 - Send PTFs Automatically (SNDPTFAUTO) attribute
 - not set correctly 273
 - Send Service Request (SNDSRVQRS)
 - command
 - description 192
 - sending more than one problem 169
 - testing communications
 - configuration 30
 - sending
 - program temporary fix 149
 - program temporary fix order 125
 - PTFs
 - automatically (SNDPTFAUTO) 123
 - Create Program Temporary Fix Package (CRPTFPKG)
 - command 136
 - do not arrive at service requester 274
 - electronically 123
 - orders to IBM service support (*IBMSRV) 119
 - to multiple service requesters 133
 - response for a problem 166, 172
 - service request 30, 169
 - test request 30
 - sequence
 - AS/400 Query display 130
 - serial number 21
 - server (NETSERVER) attribute 11
 - service information
 - for product users 150
 - for the product owner 141
 - product owner 141
 - providing 150
 - service link
 - electronic customer support 25
 - delivering PTFs 118
 - QESLINE 8
 - service number parameter
 - setting 162
 - service provider 18
 - adding
 - entries 27
 - information to PTF database 25
 - language feature support 24
 - product support 23
 - service requester entries 21, 244
 - analyzing problems 164
 - answering problems
 - hardware 165, 192
 - introduction 188
 - software 165, 189
 - attributes
 - changing 18
 - definitions 18
 - change management overview 115
 - changing attributes 18
 - configuring secondary system name
 - example 243
- service provider (*continued*)
 - creating
 - command authorization list 17
 - problem records 164
 - routing table entries 236, 243
 - deleting APAR data 167
 - displaying APAR data 167
 - example 31
 - installing System Manager 16
 - log remote problems (LOGRMTPRB)
 - setting 153
 - LOGRMTPRB (log remote problems)
 - setting 153
 - matching machine information on
 - service requester 21, 275
 - more than one 27
 - network attributes for alerts 32
 - preparing for setup 1
 - problem handling
 - filtering 176
 - grouping 176
 - receiving PTFs from 122
 - release level considerations 22
 - reporting
 - problems 165, 192
 - restrictions 193
 - routing table entry
 - creating 236, 243
 - running
 - unattended problem analysis 11, 184
 - saving APAR
 - data 166
 - library 167
 - searching for PTFs
 - matching problem symptoms 188
 - problem in ANSWERED
 - status 172
 - security
 - introduction 17
 - location password 18
 - remote location list 18
 - sending responses to the service requester 172
 - setting up
 - attributes 18
 - communications example 240, 241
 - database of PTF information 22, 275
 - introduction 14
 - list of service providers 19
 - list of service requester servers 19
 - list of service requesters 244
 - naming network systems 8
 - product support 22
 - tracking problems 175
 - working with 34, 250
 - working with entries 27
 - working with problems
 - analyze problem option 164
 - answer problem option 165
 - close problem option 166
 - delete APAR data option 167
 - display APAR data option 167
 - example 257
 - filtering 176

- service provider (*continued*)
 - working with problems (*continued*)
 - options 164
 - recovery procedures option 165
 - report problem option 165
 - save APAR data option 166
 - save APAR library option 167
 - send response option 166
 - verification procedures
 - option 165
 - Work with Problem (WRKPRB)
 - command 157
 - WRKPRB (Work with Problem)
 - command 157
 - working with PTF information 165
- service provider attribute
 - changing 18, 272
 - displaying 19, 345
 - examples 257
 - HLDPTF (hold PTFs received)
 - definition 19
 - hold PTFs received (HLDPTF)
 - definition 19
 - log remote problems (LOGRMTPRB)
 - definition 18
 - example 257, 269
 - not set correctly 272
 - setting 153
 - LOGRMTPRB (log remote problems)
 - definition 18
 - example 257, 269
 - MAXDST (maximum distribution)
 - definition 19
 - example 270
 - maximum distribution (MAXDST)
 - definition 19
 - example 270
 - message queue (MSGQ)
 - definition 19
 - example 257, 269
 - MSGQ (message queue)
 - definition 19
 - example 257, 269
 - send PTFs automatically (SNDPTFAUTO)
 - definition 19
 - example 259, 270
 - SNDPTFAUTO (send PTFs automatically)
 - definition 19
 - example 259, 270
- service provider entry
 - adding 27, 250
 - changing 28
 - copying 28, 29
 - creating 27, 250
 - for IBM service support (*IBMSRV) 28
- service request
 - problem notification
 - definition 183
 - sending 30, 169
- service requester 252
 - adding
 - service provider entries 250
 - alerts
 - filtering 13
- service requester (*continued*)
 - alerts (*continued*)
 - grouping 13
 - closing problem records 166
 - configuring
 - secondary system name 250
 - secondary system name table 250
 - configuring SNADS 235
 - copying
 - service provider 29
 - creating
 - routing table entries 236, 243
 - database of PTF information 22, 275
 - deleting APAR data 167
 - directory entries example 251
 - displaying APAR data 167
 - matching machine information 275
 - not entitled to service 275
 - preparing for setup 1
 - problem status changes 175
 - routing table entry
 - creating 236, 249
 - running problem analysis
 - unattended 183
 - when problem in SENT
 - status 186
 - saving APAR
 - data 166
 - library 167
 - secondary system name
 - verifying configuration 236
 - setting up 26, 250
 - communications 248
 - communications example 247
 - supporting prior releases 5
 - tracking problems 175
 - verifying configuration
 - secondary system name 236
 - working with 33, 273, 275
- service requester entries
 - adding 20, 244
 - changing 20
 - copying 20, 21
 - creating 244
 - deleting 20
 - displaying 20
 - matching information on the service provider 275
- service requester servers
 - example 31
- service requester systems
 - adding
 - service requester information 20
 - configuring SNADS 235
 - error messages for setup
 - verification 31
 - granting authority
 - commands running remote
 - analysis 30
 - locating machine information 21
 - OS/400 program release
 - requirements 5
 - security 29
 - status messages 30
 - verifying
 - entitlement to service 31
 - setup procedures 30
- service requests
 - problem notification
 - creating problem record 153
- servicing your products
 - using System Manager 141
- setting
 - group assigned parameter 162
 - problem severity 161
 - service number parameter 162
 - user-assigned parameter 162
- setting up
 - alerts
 - basic support 11
 - authority for remote analysis 29
 - command map 6
 - communications
 - configuration 248
 - configuration example 241
 - service provider example 240
 - service requester example 247
 - controller description
 - service provider example 241
 - service requester example 248
 - distribution feedback messages 252
 - line descriptions 241, 248
 - mapping the steps 6
 - network 1
 - parameter values map for
 - matching 239
 - service provider
 - attributes 18
 - communications example 240
 - database of PTF information 22, 275
 - introduction 14
 - list of service providers 19
 - list of service requester servers 19
 - list of service requesters 244
 - product support 22
 - setting up
 - service provider 240
 - SNA distribution services (SNADS)
 - communications for PTF
 - distribution 235
 - SNADS (SNA distribution services)
 - communications for PTF
 - distribution 235
 - System Manager 1
- setup
 - verifying 30
- severity
 - problem 161
- SNA distribution services (SNADS) 235
 - definition 235
 - directory entry 235
 - distribution queues 237, 242
 - queue
 - service provider example 242
 - requirements for System Manager 237
 - secondary system name table 237
 - setting up 242, 249
 - setting up communications for PTF
 - distribution 235
 - verifying configuration 237

- SNA Distribution Services (SNADS)
 - problem with distribution queues 274
 - problem with routing 274
 - problems with configuration 274
 - routing error 274
- SNADS
 - See SNA distribution services (SNADS)
- SNDPTF (Send Program Temporary Fix)
 - command
 - description 133
 - object authority required 345
 - sending PTFs to service requester 149
- SNDPTF (Send PTF) command
 - authorized IBM-supplied user profiles 344
- SNDPTFAUTO (send PTFs automatically)
 - attribute
 - controlling PTF distribution 123
 - definition 19
 - example 259, 270
 - releasing PTFs 132
 - setting 18
 - setting for manual intervention 22
- SNDPTFAUTO (Send PTFs Automatically) attribute
 - not set correctly 273
- SNDPTFORD (Send Program Temporary Fix Order) command
 - description 116
 - example 262
 - ordering cover letters 125
- SNDSRVRQS (Send Service Request)
 - command
 - description 192
 - sending more than one problem 169
- SNDSRVRQS (Send Service Request)
 - command 192
 - testing communications
 - configuration 30
- special PTF notification
 - defective PTFs 120
 - high-impact pervasive (HIPER) notifications 120
 - HIPER (high-impact pervasive) PTFs 120
- sphere of control
 - adding 12
 - adding to 12
 - definition 11
 - entries 12
 - changing 12
 - displaying 12
 - removing 12
 - working with 12, 35
- spooled files
 - displaying 156
- standard names 203
- Start Copy Screen (STRCPYSCN)
 - command 260
- Start Subsystem (STRSBS) command 274
- start time 201
 - using 201
- starting
 - copy screen 260
 - starting (*continued*)
 - subsystem 274
 - starting remote
 - problem analysis
 - from message queue 185
 - from problem log 185
 - from Work with Alerts display 185
 - system-detected problems 183
 - user-detected problems 186
- status
 - change request 210
- changes
 - ANSWERED 171
 - CLOSED 173
 - OPENED 167
 - PREPARED 169, 186
 - READY 168
 - SENT 170
 - VERIFIED 172
- of PTF
 - description 191
 - displaying 128
 - when answering problems 166
- problem
 - at ANSWERED 171
 - at CLOSED 173
 - at OPENED 167
 - at PREPARED 169
 - at READY 168
 - at SENT 170
 - at VERIFIED 172
- problem records
 - SENT 170
- table of changes 174, 175
- status change 167
 - performing operations
 - summary 173
- PREPARED status 186
- problem records
 - ANSWERED 171
 - CLOSED 173
 - OPENED status 167
 - PREPARED 169
 - READY 168
 - VERIFIED 172
- storage space
 - conserving 12
 - managing 6
 - planning 6
- storing PTFs 6, 121
- strategy, license term 94
- STRCPYSCN (Start Copy Screen)
 - command 260
- stream files
 - directories
 - PTF 146
- submitted change request 210, 211
 - message details 216
 - messages, working with 215
 - working with activities 212
- submitting
 - change request 208
- subsystem
 - QSNADS
 - ending 28
 - not active 274
- subsystem (*continued*)
 - starting 274
- superseded PTF 144
- superseded relationship 143
- supported product PTFs
 - ordering 119, 121
- supported products
 - working with 33, 115, 345
- supporting
 - multiple releases of a product 42
 - product
 - installing new release 22
 - supporting more than one release 22
 - system at different release level 4
- symptom string
 - adding to the PTF 145
- syntax diagram 233
- system
 - DSPSYSVAL (Display System Value)
 - command 21
 - map for setup 6
 - naming conventions 8
 - reporting problems detected by 182
 - serial number 21
 - verifying names 10
- System Manager licensed program
 - correcting problems 272
 - servicing your products 141
 - setting up 1
- SNA distribution services (SNADS)
 - requirements 237
- SNADS (SNA distribution services)
 - requirements 237
- troubleshooting 272
- verifying system names
 - LCLCPNAME (local control name) 10
 - LCLLOCNAME (local location name) 10
 - Local control point name (LCLCPNAME) 10
 - Local location name (LCLLOCNAME) 10
 - SYSNAME (system name) 10
 - system name (SYSNAME) 10
- system serial number (QSRLNBR) system value 21
- system value
 - automatic virtual device configuration (QAUTOVRT)
 - changing 30, 253
 - definition 29
 - not set correctly 275
 - changing 253, 275
 - displaying 275
 - QAUTOVRT (automatic virtual device configuration)
 - changing 30, 253
 - definition 29
 - not set correctly 275
 - QPRBHLDTIV (problem hold interval) 163
 - QRMTSIGN (remote sign-on verification)
 - changing 30, 253
 - definition 29

- system value (*continued*)
 - QRMTSIGN (remote sign-on verification) (*continued*)
 - not set correctly 275
 - QSRLNBR (system serial number) 21
 - remote sign-on verification (QRMTSIGN)
 - changing 30
 - definition 29
 - not set correctly 275
 - system serial number (QSRLNBR) 21
- system-detected problem 180

T

- table
 - problem status changes 174, 175
 - release-to-release support 5
- tape
 - delivering PTFs 118
 - verifying 195
- test request
 - *TEST parameter for communications 30
 - problems with 31
 - sending 30
- testing
 - installed products 112
 - PTF 148
- time zone
 - local 201
 - remote 202
 - values 201
- time zone values 201
- tips and techniques
 - packaging product 86
- tracking
 - change requests 210
 - problem management operations 153
 - problems 175
- transferring PTFs
 - from tape
 - download 121
- translating
 - language load 53
 - language loads for your application 67
 - language loads for your product 67
 - to another language 41
- troubleshooting System Manager licensed program 272
- type of compliance regulations to enforce 95

U

- UNATTEND network attribute value 11
- usage types
 - concurrent 95
 - registered 95
- user profile
 - basic service representative (QSRVBAS) 236
 - creating 29, 252
 - creating for remote problem analysis 252

- user profile (*continued*)
 - database file 355
 - database logical file 355
 - not correct for remote problem analysis 275
 - QSECOFR (security officer) 29
 - QSRVBAS (basic security representative) 123
 - QSRVBAS (basic service representative) 236
 - QSVSM authorities 369
 - QSYSOPR (system operator) 29
 - security officer (QSECOFR) 29
 - system operator (QSYSOPR) 29
- user profile database file 355
- user-assigned parameter
 - setting 162
- user-based pricing
 - ordering process changes 94
 - packaging products 93
- user-detected problems 181
- using
 - commands on managed systems 229
 - using object distribution for sending PTF 122
 - using output file 159
 - writing programs 128
 - using output files 159
 - using System Manager licensed program servicing your products 141

V

- verification
 - missing folders 76
 - missing objects 76
- Verify Communications (VFYCMN) command 195
- Verify Tape (VFYTAP) command 195
- verifying
 - communications 195
 - contact information 14
 - problem correction 165
 - service requesters entitled to service 31
 - setup procedures 30
 - SNADS configuration 237
 - system name 10
 - tape 195
- VFYCMN (Verify Communications) command 195
- VFYTAP (Verify Tape) command 195
- vital product data
 - definition 166
 - information in problem record 180

W

- work sheet
 - filled-in example 254
- Work with Active Jobs (WRKACTJOB) command 260
- Work with Alerts (WRKALR) command
 - description 194
 - displaying alerts 12
 - example 270

- Work with Alerts (WRKALR) command (*continued*)
 - starting remote problem analysis 185
- Work with Authorization Lists (WRKAUTL) command 17
- Work with Configuration List (WRKCFGL) command 18
- Work with Contact Information (WRKCNTINF) command
 - changing cover letter language 125
 - example 271
 - verifying service information 14
- Work with Controller Descriptions (WRKCTLD) command 12
- Work with Distribution Queue (WRKDSTQ) command 274
- Work with License Information (WRKLICINF) command 94, 103
- Work with problem (WRKPRB) command
 - description 194
- Work with Problem (WRKPRB) command 259
 - answering software problems 189
 - authorization to 29, 275
 - changing problem status to run local analysis 186
 - description 157
 - displaying problem details 176
 - examples 158
 - starting remote problem analysis 185, 187
- Work with Program Temporary Fix (WRKPTF) command
 - description 127
 - object authority required 345
 - viewing PTF attributes 142
- Work with Service Providers (WRKSRVPVD) command
 - adding entries 27
 - changing route to IBM service support 34
 - example 250, 252
- Work with Service Requesters (WRKSRVRQS) command
 - adding entries 19
 - authorized IBM-supplied user profiles 344
 - correcting machine information 275
 - determining default message queue 273
 - example 33, 244
 - object authority required 345
 - verifying
 - attribute values 273
 - information 272
- Work with Sphere of Control (WRKSOC) command 35
 - using 12
- Work with Supported Products (WRKSPTPRD) command
 - adding product support 275
 - authorized IBM-supplied user profiles 344
 - description 22, 115
 - example 33, 262
 - object authority required 345
 - verifying product support 273

- working with
 - active job 260
 - activities 212
 - alerts 12, 185, 270
 - filtering 13
 - grouping 13
 - authorization lists 17
 - change request activities 212
 - change request activity details 205
 - configuration lists 18
 - contact information 125
 - example 271
 - controller descriptions 12
 - cover letters 125
 - distribution queue 274
 - license information 94, 103
 - problem 157, 259
 - displaying details 176
 - problem log 153, 182
 - problems
 - filtering 176
 - grouping 176
 - options 164
 - program temporary fix 345
 - program temporary fixes 142
 - PTF information 165
 - PTF save files 124
 - service provider 250
 - service providers 34
 - service requester servers
 - adding entries 19
 - service requesters 33, 273, 275
 - sphere of control 12, 35
 - submitted change request messages 215
 - supported products 33, 115, 345
- writing program 159
 - using output files 128
- writing programs 159
- WRKACTJOB (Work with Active Jobs) command 260
- WRKALR (Work with Alerts) command
 - description 194
 - displaying alerts 12
 - example 270
 - starting remote problem analysis 185
- WRKAUTL (Work with Authorization Lists) command 17
- WRKCFGL (Work with Configuration List) command 18
- WRKCNTINF (Work with Contact Information) command
 - changing cover letter language 125
 - example 271
 - verifying service information 14
- WRKCTLD (Work with Controller Descriptions) command 12
- WRKDSTQ (Work with Distribution Queue) command 274
- WRKLCINF (Work with License Information) command 94, 103
- WRKPRB (Work with Problem) command
 - answering software problems 189
 - authorization to 29
 - changing problem status to run local analysis 186
 - description 194
- WRKPRB (Work with Problem)
 - command (*continued*)
 - command description 157
 - Work with Problem (WRKPRB) 157
 - WRKPRB (Work with Problem) command 157
 - displaying problem details 176
 - examples 158, 259
 - starting remote problem analysis 185, 187
- WRKPTF (Work with Program Temporary Fix) command
 - authorized IBM-supplied user profiles 344
 - description 127
 - object authority required 345
 - viewing PTF attributes 142
- WRKSOC (Work with Sphere of Control)
 - command 12, 35
 - using 12
- WRKSPTPRD (Work with Supported Products) command
 - adding product support 275
 - authorized IBM-supplied user profiles 344
 - description 22, 115
 - example 33, 262
 - object authority required 345
 - verifying product support 273
- WRKSRVPVD (Work with Service Providers) command
 - adding entries 27
 - changing route to IBM service support 34
 - example 250, 252
- WRKSRVRQS (Work with Service Requesters) command
 - adding entries 19
 - authorized IBM-supplied user profiles 344
 - correcting machine information 275
 - determining default message queue 273
 - example 33, 244
 - object authority required 345
 - verifying
 - attribute values 273
 - information 272

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

iSeries
System Manager Use
Version 5

Publication No. SC41-5321-02

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

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

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

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

Please tell us how we can improve this book:

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

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

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

IBM CORPORATION
ATTN DEPT 542 IDCLERK
3605 HWY 52 N
ROCHESTER MN 55901-7829



Fold and Tape

Please do not staple

Fold and Tape



Printed in U.S.A.

SC41-5321-02

