

AS/400 Advanced Series



# System/36 Environment Reference

*Version 3*



AS/400 Advanced Series



# System/36 Environment Reference

*Version 3*

**Take Note!**

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

### **First Edition (September 1995)**

This edition applies to the licensed program IBM Operating System/400, (Program 5716-SS1), Version 3 Release 6 Modification 0, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the proper edition for the level of the product.

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

A form for reader comments is provided at the back of this publication. If the form has been removed, you can mail your comments to:

Attn Department 542  
IDCLERK  
IBM Corporation  
3605 Highway 52 N  
Rochester, MN 55901-9986 USA

or you can fax your comments to:

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

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

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you or restricting your use of it.

© **Copyright International Business Machines Corporation 1995. All rights reserved.**

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

---

# Contents

<b>Notices</b> . . . . .	xvii
Trademarks And Service Marks . . . . .	xvii
<b>About System/36 Environment Reference (SC41-4731)</b> . . . . .	xix
Who Should Use This Book . . . . .	xix
<b>Chapter 1. Introduction</b> . . . . .	1-1
Procedures . . . . .	1-1
OCL Statements . . . . .	1-2
System/36 Environment Utility Programs and Their Control Statements . . . . .	1-2
Control Commands . . . . .	1-3
Concept Information and Programming Considerations . . . . .	1-3
Conventions Used for Describing Syntax Formats . . . . .	1-3
Capitalized Expressions . . . . .	1-3
Braces { } . . . . .	1-3
Brackets [ ] . . . . .	1-4
Parentheses ( ) . . . . .	1-4
Underscores _ . . . . .	1-4
Commas , . . . . .	1-5
Example Syntax Diagram . . . . .	1-5
Directory to Using System Support . . . . .	1-6
Creating and Maintaining Disk Files . . . . .	1-6
Creating and Maintaining Extended Character Files . . . . .	1-7
Creating and Maintaining Libraries . . . . .	1-8
Maintaining Folders and Folder Members . . . . .	1-9
Processing Diskettes . . . . .	1-9
Processing Tapes . . . . .	1-10
Creating and Maintaining Display Formats, Menus, and Message Members . . . . .	1-11
Creating and Maintaining Programs . . . . .	1-11
Using the Office Products . . . . .	1-12
Using the Personal Computer . . . . .	1-13
Defining Data in Files . . . . .	1-13
Running Programs and Procedures . . . . .	1-13
Changing and Controlling Printers, Jobs, and Display Stations . . . . .	1-15
Starting Procedures Automatically . . . . .	1-17
Communicating with Other Systems . . . . .	1-17
Maintaining the System . . . . .	1-17
Determining and Correcting Problems . . . . .	1-17
Reference Numbers for Library Members . . . . .	1-18
<b>Chapter 2. Making Your Own Procedures</b> . . . . .	2-1
What a Procedure Is . . . . .	2-1
What a Procedure Can Contain . . . . .	2-1
Entering Procedures into the System . . . . .	2-2
Naming Procedures . . . . .	2-2
Procedure Parameters . . . . .	2-3
Procedure Parameter Defaults . . . . .	2-5
Testing Entered Parameters . . . . .	2-6
Parameter Coding Considerations . . . . .	2-6
Continuing the Lines of a Procedure . . . . .	2-7

Calling a Procedure from Another Procedure . . . . .	2-8
Procedure Attributes . . . . .	2-9
Example Procedures . . . . .	2-10
Example 1: Procedure SAMPLE . . . . .	2-10
Example 2: Procedure LISTKEYS . . . . .	2-11
Example 3: Procedure SCRNPRT . . . . .	2-14
Control Language Commands in Procedures . . . . .	2-20
Using Substitution Expressions on CL Commands . . . . .	2-20
Prompting for CL Commands . . . . .	2-22
Conditioning CL Commands . . . . .	2-22
Special Characters in Names . . . . .	2-22
Handling Errors on CL Commands . . . . .	2-22
Procedure Performance Tips and Coding Techniques . . . . .	2-24
Debugging Your Procedures . . . . .	2-27
<b>Chapter 3. Procedure Control Expressions . . . . .</b>	<b>3-1</b>
Things You Can Do Using Procedure Control Expressions . . . . .	3-1
Substituting Values and Information . . . . .	3-1
Displaying Messages or Display Formats . . . . .	3-3
Data File Information . . . . .	3-4
Library Information . . . . .	3-4
Diskette Information . . . . .	3-5
Tape Information . . . . .	3-5
Comparing and Evaluating Values and Branching in Procedures . . . . .	3-5
Testing the Procedure or Job Environment . . . . .	3-6
Ending Procedures . . . . .	3-6
* (Comment) Statement . . . . .	3-7
Substitution Expressions . . . . .	3-7
?n? (Parameter) Expression . . . . .	3-8
?n'value'? (Default Parameter) Expression . . . . .	3-9
?nT'value'? (Temporary Value Parameter) Expression . . . . .	3-9
?nF'value'? (Forced Value Parameter) Expression . . . . .	3-10
?R? (Required Parameter) Expression . . . . .	3-10
?nR? (Missing Parameter) Expression . . . . .	3-11
?R'mic'? (Required Parameter Message) Expression . . . . .	3-11
?nR'mic'? (Missing Parameter Message) Expression . . . . .	3-12
?Cn? (Parameter Length) Expression . . . . .	3-12
?C'value'? (Length) Expression . . . . .	3-13
?CD? (Return Code) Expression . . . . .	3-13
?CLIB? (Current Library) Expression . . . . .	3-16
?DATE? (Program Date) Expression . . . . .	3-16
?DATEFMT? (Date Format) Expression . . . . .	3-17
?DEV'unit'? (Device Name) Expression . . . . .	3-17
?F'S,name'? or ?F'S,name,date'? (File Size) Expression . . . . .	3-17
?F'A,name'? or ?F'A,name,date'? (Actual File Size) Expression . . . . .	3-18
?F'T,name'? (File Type) Expression . . . . .	3-19
?FLIB? (Files Library) Expression . . . . .	3-19
?L'position,length'? (Local Data Area) Expression . . . . .	3-19
?Mmic? or ?M'mic,position,length'? (Message Member) Expression . . . . .	3-20
?MENU? (Current Menu) Expression . . . . .	3-21
?MSGID? (Message ID) Expression . . . . .	3-21
?PRINTER? (Session Printer) Expression . . . . .	3-21
?PROC? (First Level Procedure) Expression . . . . .	3-22
?SFLIB? (Session Files Library) Expression . . . . .	3-22

?SLIB? (Session Library) Expression . . . . .	3-22
?SYSLIST? (System List Device) Expression . . . . .	3-22
?TIME? (System Time) Expression . . . . .	3-23
?USER? (Operator's User ID) Expression . . . . .	3-23
?VOLID? or ?VOLID'location'? (Diskette or Tape Volume ID) Expression . . . . .	3-23
?WS? (Display Station ID) Expression . . . . .	3-24
Nested Substitution Expressions . . . . .	3-24
IF Conditional Expressions . . . . .	3-25
ACTIVE (Running Procedures) Condition . . . . .	3-26
BLOCKS (Available Disk Space) Condition . . . . .	3-27
CONSOLE (System Console) Condition . . . . .	3-27
DATAF1 (Files and Libraries on Disk) Condition . . . . .	3-28
DATAI1 (Files on Diskette) Condition . . . . .	3-29
DATAT (Files on Tape) Condition . . . . .	3-30
DSPLY (Display Station Type) Condition . . . . .	3-31
ENABLED (Enabled Communications) Condition . . . . .	3-32
EVOKED (Evoked Procedures) Condition . . . . .	3-32
FILELIBL (File Library List) Condition . . . . .	3-32
INQUIRY (Inquiry Mode) Condition . . . . .	3-33
JOBQ (Job Queue) Condition . . . . .	3-33
LISTDONE (Phone List Completion) Condition . . . . .	3-34
LOAD (Library Load Members) Condition . . . . .	3-34
MRTMAX (Multiple Requesting Terminals) Condition . . . . .	3-35
PROC (Library Procedure Members) Condition . . . . .	3-36
SECURITY (Password Security) Condition . . . . .	3-36
SOURCE (Library Source Members) Condition . . . . .	3-37
SUBR (Library Subroutine Members) Condition . . . . .	3-38
SWITCH (Switches) Condition . . . . .	3-39
SWITCHn (Individual Switches) Condition . . . . .	3-39
string1=string2 (Comparing, Equal To) Condition . . . . .	3-40
string1>string2 (Comparing, Greater Than) Condition . . . . .	3-41
VOLID (Diskette and Tape Volume IDs) Condition . . . . .	3-41
Statement Portion of IF Conditional Expression . . . . .	3-43
ELSE Expressions . . . . .	3-43
// * (Informational Message) Statement . . . . .	3-44
// ** (System Console Message) Statement . . . . .	3-45
CANCEL Statement . . . . .	3-46
EVALUATE Statement . . . . .	3-46
Assigning Values . . . . .	3-47
Adding, Subtracting, Multiplying, or Dividing . . . . .	3-48
Evaluating Substitution Expressions . . . . .	3-51
Setting the Return Code . . . . .	3-51
GOTO and TAG Statements . . . . .	3-52
PAUSE Statement . . . . .	3-53
RESET Statement . . . . .	3-54
RETURN Statement . . . . .	3-55
<b>Chapter 4. Procedures . . . . .</b>	<b>4-1</b>
Managing Licensed Programs . . . . .	4-2
#STRUP1 Procedure . . . . .	4-3
#STRUP2 Procedure . . . . .	4-3
ALERT Procedure . . . . .	4-4
ALOCFLDR Procedure . . . . .	4-4
ALOCLIBR Procedure . . . . .	4-5

ALTERBSC Procedure . . . . .	4-5
ALTERCOM Procedure . . . . .	4-5
ALTERSDL Procedure . . . . .	4-10
APAR Procedure . . . . .	4-10
APPNINFO Procedure . . . . .	4-10
ARCHIVE Procedure . . . . .	4-11
ASM Procedure . . . . .	4-11
ASMLOAD Procedure . . . . .	4-11
ASMSAVE Procedure . . . . .	4-11
AUTO Procedure . . . . .	4-11
AUTO C Procedure . . . . .	4-11
BACKUP Procedure . . . . .	4-14
BALPRINT Procedure . . . . .	4-14
BASIC Procedure . . . . .	4-14
BASICP Procedure . . . . .	4-15
BASICR Procedure . . . . .	4-15
BASICS Procedure . . . . .	4-15
BASLOAD Procedure . . . . .	4-15
BASSAVE Procedure . . . . .	4-15
BGUATTR Procedure . . . . .	4-16
BGUCHART Procedure . . . . .	4-16
BGUDATA Procedure . . . . .	4-16
BGUGRAPH Procedure . . . . .	4-16
BGULOAD Procedure . . . . .	4-17
BGUSAVE Procedure . . . . .	4-17
BLDFILE Procedure . . . . .	4-17
BLDGRAPH Procedure . . . . .	4-19
BLDINDEX Procedure . . . . .	4-20
BLDLIBR Procedure . . . . .	4-22
BLDMENU Procedure . . . . .	4-25
BUILD Procedure . . . . .	4-29
CACHE Procedure . . . . .	4-29
CATALOG Procedure . . . . .	4-29
Sample Disk VTOC Listings . . . . .	4-33
Diskette VTOC Listing . . . . .	4-39
Tape Label Listings . . . . .	4-39
CGU Procedure . . . . .	4-39
CGULOAD Procedure . . . . .	4-39
CGUSAVE Procedure . . . . .	4-39
CHGXLATE Procedure . . . . .	4-40
CHNGEMEM Procedure . . . . .	4-40
CLRPF Procedure . . . . .	4-43
CNFIGICF Procedure . . . . .	4-43
CNFIGSSP Procedure . . . . .	4-43
CNFIGX25 Procedure . . . . .	4-44
COBLOAD Procedure . . . . .	4-44
COBOL Procedure . . . . .	4-44
COBOLC Procedure . . . . .	4-44
COBOLCG Procedure . . . . .	4-47
COBOLG Procedure . . . . .	4-47
COBOLONL Procedure . . . . .	4-47
COBOLP Procedure . . . . .	4-47
COBSAVE Procedure . . . . .	4-48
COBSDA Procedure . . . . .	4-48

COBSEU Procedure . . . . .	4-48
COMPRESS Procedure . . . . .	4-49
CONDENSE Procedure . . . . .	4-49
COPYDATA Procedure . . . . .	4-50
COPYDIAG Procedure . . . . .	4-55
COPYI1 Procedure . . . . .	4-55
COPYPRT Procedure . . . . .	4-57
File Format for COPYPRT Files . . . . .	4-59
CREATE Procedure . . . . .	4-66
Message Member Statements . . . . .	4-67
Considerations for the Double-Byte Character Set Version of the OS/400 Program . . . . .	4-70
CSALL Procedure . . . . .	4-71
DATE Procedure . . . . .	4-71
DEFINEID Procedure . . . . .	4-73
DEFINEPN Procedure . . . . .	4-73
DEFINLOC Procedure . . . . .	4-74
DEFINX21 Procedure . . . . .	4-74
DEFINX25 Procedure . . . . .	4-75
DEFSUBD Procedure . . . . .	4-75
DELETE Procedure . . . . .	4-76
DELNRD Procedure . . . . .	4-80
DFA Procedure . . . . .	4-80
DFU Procedure . . . . .	4-80
DFULOAD Procedure . . . . .	4-81
DFUSAVE Procedure . . . . .	4-81
DICTLOAD Procedure . . . . .	4-81
DICTSAVE Procedure . . . . .	4-81
DISABLE Procedure . . . . .	4-81
DISPLAY Procedure . . . . .	4-82
DLSLOAD Procedure . . . . .	4-82
DLSSAVE Procedure . . . . .	4-82
DOCCNV Procedure . . . . .	4-82
DOCPLOAD Procedure . . . . .	4-82
DOCPSAVE Procedure . . . . .	4-82
DSPLOCKS Procedure . . . . .	4-83
DSPMNU Procedure . . . . .	4-83
DSPSYS Procedure . . . . .	4-83
DSU Procedure . . . . .	4-83
DSULOAD Procedure . . . . .	4-85
DSUSAVE Procedure . . . . .	4-85
DUMP Procedure . . . . .	4-86
EDITNRD Procedure . . . . .	4-86
EM3270 Procedure . . . . .	4-86
ENABLE Procedure . . . . .	4-89
ENTER Procedure . . . . .	4-89
ENTER# Procedure . . . . .	4-91
EPDOWNL Procedure . . . . .	4-92
EPLMRG Procedure . . . . .	4-92
EP3270 Procedure . . . . .	4-92
ERAP Procedure . . . . .	4-93
ERR Procedure . . . . .	4-93
ES3270 Procedure . . . . .	4-95
EXTRACT Procedure . . . . .	4-97

FLIB Procedure	4-98
FORMAT Procedure	4-99
FORTRAN Procedures	4-103
FROMLIBR Procedure	4-103
HELP Procedure	4-109
HISTCOPY Procedure	4-114
HISTCRT Procedure	4-114
HISTORY Procedure	4-115
ICFDEBUG Procedure	4-115
ICVERIFY Procedure	4-115
IDDU Procedure	4-115
IDDUDCT Procedure	4-116
IDDUDFN Procedure	4-116
IDDUDISK Procedure	4-116
IDDULINK Procedure	4-116
IDDUPRT Procedure	4-117
IDDURBLD Procedure	4-118
IDDUXLAT Procedure	4-119
IGC Procedure	4-119
INIT Procedure	4-119
INITDIAG Procedure	4-121
INIT9332 Procedure	4-121
INQUIRY Procedure	4-121
INQUIRY# Procedure	4-123
IPL Procedure	4-124
ITF Procedure	4-124
IWLOAD Procedure	4-125
IWPTLOAD Procedure	4-125
IWPTSAVE Procedure	4-125
IWSAVE Procedure	4-125
JOBSTR Procedure	4-125
KEYS Procedure	4-128
KEYSORT Procedure	4-129
LANLOAD Procedure	4-129
LANSAVE Procedure	4-129
LIBRLIBR Procedure	4-129
LINES Procedure	4-132
LIST Procedure	4-134
LIST# Procedure	4-135
LISTDATA Procedure	4-137
LISTFILE Procedure	4-142
LISTLIBR Procedure	4-147
Sample LISTLIBR Listings	4-151
LISTNRD Procedure	4-154
LOAD3601 Procedure	4-155
LOG Procedure	4-155
LRTRLOAD Procedure	4-156
LRTRSAVE Procedure	4-156
MAINTX25 Procedure	4-156
MCSCONV Procedure	4-156
MOVEFLDR Procedure	4-156
MSDOWNL Procedure	4-157
MS3270 Procedure	4-157
MSGFILE Procedure	4-158



REMOVE Procedure	4-192
RENAME Procedure	4-194
REQUESTX Procedure	4-195
RESPONSE Procedure	4-196
Automatic Response Source Statements	4-197
Automatic Response Programming Considerations	4-201
RETEXTN Procedure	4-202
RESTFLDR Procedure	4-203
RESTLIBR Procedure	4-203
RESTNRD Procedure	4-205
RESTORE Procedure	4-206
RETRIEVE Procedure	4-212
RGZFILE Procedure	4-212
RJFILE Procedure	4-213
RJTABLE Procedure	4-213
ROLLKEYS Procedure	4-214
RPG Procedure	4-214
RPGC Procedure	4-214
RPGLOAD Procedure	4-216
RPGONL Procedure	4-217
RPGR Procedure	4-217
RPGR Procedure	4-217
RPGSAVE Procedure	4-218
RPGSDA Procedure	4-218
RPGSEU Procedure	4-219
RPGX Procedure	4-219
SAVE Procedure	4-220
SAVEEXTN Procedure	4-227
SAVEFLDR Procedure	4-228
SAVELIBR Procedure	4-228
SAVENRD Procedure	4-230
SDA Procedure	4-230
SDALOAD Procedure	4-231
SDASAVE Procedure	4-232
SECDEF Procedure	4-232
SECEDIT Procedure	4-232
SECLIST Procedure	4-233
SECRET Procedure	4-234
SECSAVE Procedure	4-234
SERVICE Procedure	4-234
SERVLOG Procedure	4-235
SET Procedure	4-235
SETALERT Procedure	4-237
SETCOMM Procedure	4-237
SETDUMP Procedure	4-242
SEU Procedure	4-242
SEULOAD Procedure	4-243
SEUSAVE Procedure	4-243
SHRFLOAD Procedure	4-244
SHRFSAVE Procedure	4-244
SLIB Procedure	4-244
SMF Procedure	4-245
SMFDATA Procedure	4-245
SMFPRINT Procedure	4-245

SMFSTART Procedure	4-246
SMFSTOP Procedure	4-246
SOFTWARE Procedure	4-246
SORT Procedure	4-247
SPECIFY Procedure	4-247
SRTX Procedure	4-248
SRTXBLD Procedure	4-249
SRTXLOAD Procedure	4-249
SRTXSAVE Procedure	4-249
STARTM Procedure	4-249
STATEST Procedure	4-250
STOPGRP Procedure	4-251
STOPM Procedure	4-251
STRTRGRP Procedure	4-251
SWDLOAD Procedure	4-252
SWDSAVE Procedure	4-252
SWITCH Procedure	4-252
SYSLIST Procedure	4-253
SYSTYPE Procedure	4-255
TAPECOPY Procedure	4-255
TAPEINIT Procedure	4-262
TAPESTAT Procedure	4-263
TEXTCONV Procedure	4-263
TEXTDCT Procedure	4-264
TEXTDOC Procedure	4-264
File Format for PRTFILE Files	4-268
TEXTFLDR Procedure	4-280
TEXTLOAD Procedure	4-280
TEXTOBJ Procedure	4-281
TEXTPROF Procedure	4-281
TEXTPRTQ Procedure	4-281
TEXTREL Procedure	4-281
TEXTSAVE Procedure	4-282
TIMER Procedure	4-282
TOLIBR Procedure	4-283
TRACE Procedure	4-288
TRANSFER Procedure	4-289
Basic Data Exchange Files	4-293
I-Exchange Files	4-293
TRNMGR Procedure	4-293
UPDATE Procedure	4-294
UPDATE# Procedure	4-295
WRKSPL Procedure	4-297
WRKUSER Procedure	4-297
WSFLOAD Procedure	4-297
WSFSAVE Procedure	4-297
WSU Procedures	4-298
XREST Procedure	4-298
XSAVE Procedure	4-298
<b>Chapter 5. OCL Statements</b>	<b>5-1</b>
Placement of OCL Statements	5-1
Types of Information Contained in OCL Statements	5-1
Identifiers	5-2

OCL Parameters	5-2
Procedure Parameters	5-2
General OCL Coding Rules	5-3
Continuing OCL Statements	5-4
Comments	5-4
ABEND OCL Statement	5-5
ALLOCATE OCL Statement	5-6
ATTR OCL Statement	5-8
CANCEL OCL Statement	5-13
CHANGE OCL Statement	5-14
COMM OCL Statement	5-16
COMPILE OCL Statement	5-17
DATE OCL Statement	5-19
DEALLOC OCL Statement	5-21
DEBUG OCL Statement	5-22
EVOKE OCL Statement	5-23
FILE OCL Statement (for Disk Files)	5-25
FILE OCL Statement (for Diskette Files)	5-33
FILE OCL Statement (for Tape Files)	5-37
FILELIB OCL Statement	5-43
FORMS OCL Statement	5-46
IMAGE OCL Statement	5-49
INCLUDE OCL Statement	5-49
INFOMSG OCL Statement	5-51
JOBQ OCL Statement	5-52
LIBRARY OCL Statement	5-54
LOAD OCL Statement	5-55
LOCAL OCL Statement	5-56
LOG OCL Statement	5-58
MEMBER OCL Statement	5-59
MENU OCL Statement	5-61
MSG OCL Statement	5-62
NOHALT OCL Statement	5-65
OFF OCL Statement	5-66
POWER OCL Statement	5-67
PRINTER OCL Statement	5-68
PROMPT OCL Statement	5-77
REGION OCL Statement	5-82
RESERVE OCL Statement	5-82
RUN OCL Statement	5-83
RUN400 OCL Statement	5-83
SESSION OCL Statement	5-84
For APPC and APPN	5-85
For SNA Upline Facility	5-86
For BSCEL and RPG II Telecommunications	5-87
For Asynchronous Communications	5-91
For Finance Communications	5-91
For Retail Communications	5-92
For Intrasystem Communications	5-93
SETDEV OCL Statement	5-94
START OCL Statement	5-95
STOP OCL Statement	5-96
SWITCH OCL Statement	5-97
SYSLIST OCL Statement	5-98

TIMERSET OCL Statement	5-100
VARY OCL Statement	5-102
WAIT OCL Statement	5-103
WORKSTN OCL Statement	5-105
/* (End of Data) Statement	5-107
<b>Chapter 6. Control Commands</b>	6-1
ASSIGN Control Command	6-3
CANCEL Control Command	6-3
CHANGE Control Command	6-5
CONSOLE Control Command	6-11
HOLD Control Command	6-11
INFOMSG Control Command	6-12
JOBQ Control Command	6-12
MENU Control Command	6-14
MODE Control Command	6-15
MSG Control Command	6-15
OFF Control Command	6-18
POWER Control Command	6-19
PRTY Control Command	6-19
RELEASE Control Command	6-19
REPLY Control Command	6-20
RESTART Control Command	6-20
START Control Command	6-22
STATUS Control Command	6-24
STATUSF Control Command	6-30
STOP Control Command	6-32
TIME Control Command	6-34
VARY Control Command	6-34
<b>Appendix A. System/36 Environment Utility Programs</b>	A-1
Making Your Own Procedures Using System/36 Environment Utility Programs	A-1
\$ARSP Utility	A-3
Changing Automatic Response Values (RESPONSE Procedure)	A-3
Changing Alert Indicators (SETALERT Procedure)	A-3
\$BICR Utility	A-3
Listing Diskette Files (LISTFILE Procedure)	A-3
Copying Files (TRANSFER Procedure)	A-4
\$BMENU Utility	A-5
\$BUILD Utility	A-6
\$COPY Utility	A-7
Copying Disk Files (COPYDATA Procedure)	A-7
Listing \$COPY Files (LISTDATA/LISTFILE Procedures)	A-11
Restoring the Network Resource Directory (RESTNRD Procedure)	A-14
Restoring Files (RESTORE Procedure)	A-14
Saving Files (SAVE Procedure)	A-19
Saving the Network Resource Directory (SAVENRD Procedure)	A-24
IBM System/34 Compatible Statements	A-24
\$CPPE Utility	A-28
Differences from the Procedure Parameters	A-28
\$CZUT Utility	A-28
\$DCOPY Utility	A-29
\$DDST Utility	A-29
\$DELET Utility	A-29

\$DPGP Utility	A-31
\$DPGR Utility	A-32
\$DUPRD Utility	A-32
\$FBLD Utility	A-33
Creating Files (BLDFILE Procedure)	A-33
Creating Alternative Indexes (BLDINDEX Procedure)	A-34
\$FREE Utility	A-34
\$HELP Utility	A-35
\$HIST Utility	A-35
\$IDSET Utility	A-35
\$IEDS Utility	A-35
\$IENBL Utility	A-35
\$INIT Utility	A-35
\$LABEL Utility	A-36
\$MAINT Utility	A-37
Create a Library (BLDLIBR Procedure)	A-37
Create Source or Procedure Members	A-39
Change Library or Directory Size (ALOCLIBR Procedure)	A-42
Change Library Member Information (CHNGEMEM Procedure)	A-43
Gather Unused Library Space (CONDENSE Procedure)	A-43
Copy Members from One Library to Another (LIBRLIBR Procedure)	A-44
Copy Members from a Library (FROMLIBR Procedure)	A-45
COPY and CEND Statements	A-47
Copy Members to a Library (TOLIBR Procedure)	A-50
Start a Job (JOBSTR Procedure)	A-51
List Library Members and Information (LISTLIBR Procedure)	A-51
List Information about Libraries (LISTFILE Procedure)	A-52
Remove Members from a Library (REMOVE Procedure)	A-53
Save a Library (SAVELIBR Procedure)	A-54
Restore a Library (RESTLIBR Procedure)	A-55
\$MGBLD Utility	A-55
\$MMSP Utility	A-56
\$MMST Utility	A-56
\$PACK Utility	A-56
\$PNLM Utility	A-56
\$POST Utility	A-57
\$PRCED Utility	A-57
\$PRCLT Utility	A-57
\$PRLST Utility	A-57
\$PRPWD Utility	A-57
\$PRUED Utility	A-57
\$PRUID Utility	A-57
\$PRURS Utility	A-57
\$PRUSV Utility	A-58
\$RENAM Utility	A-58
\$RREDT Utility	A-58
\$RRESC Utility	A-58
\$RRLST Utility	A-58
\$RRSAV Utility	A-59
\$RRSTR Utility	A-59
\$RRTED Utility	A-59
\$RRTLT Utility	A-59
\$SETCF Utility	A-59
Change Communications Attributes (ALTERCOM Procedure)	A-59

Set Display Station Environment (SET Procedure) . . . . .	A-59
Specify Print Key Information (PRINTKEY Procedure) . . . . .	A-59
\$SETCP Utility . . . . .	A-60
\$SFGR Utility . . . . .	A-60
\$SINCT Utility . . . . .	A-62
\$SINDL Utility . . . . .	A-62
\$SINR Utility . . . . .	A-62
\$SVCASRV Utility . . . . .	A-62
\$TCOPY Utility . . . . .	A-62
Copy Data To or From Tape (TAPECOPY Procedure) . . . . .	A-63
List the Contents of Exchange Tape Files (LISTFILE Procedure) . . . . .	A-63
\$TINIT Utility . . . . .	A-64
\$TMSERV Utility . . . . .	A-64
\$UASC Utility . . . . .	A-65
\$UASF Utility . . . . .	A-65
\$XNLM Utility . . . . .	A-66
\$XNSH Utility . . . . .	A-66
\$XREST Utility . . . . .	A-66
\$XSAVE Utility . . . . .	A-66
#GCFR Utility . . . . .	A-66
#GSORT Utility . . . . .	A-66
File Information Needed When Writing OCL Statements . . . . .	A-68
#KASRT Utility . . . . .	A-69
<b>Appendix B. Converting Decimal and Hexadecimal Numbers</b> . . . . .	B-1
Converting Decimal to Hexadecimal . . . . .	B-1
Hexadecimal to Decimal Example . . . . .	B-2
Decimal to Hexadecimal Example . . . . .	B-2
Records to Blocks Conversion for Disk Files . . . . .	B-3
Determining the Number of Blocks in a File . . . . .	B-3
<b>Appendix C. EBCDIC and ASCII Code Tables</b> . . . . .	C-1
EBCDIC . . . . .	C-1
ASCII . . . . .	C-2
<b>Appendix D. SFGR System Messages</b> . . . . .	D-1
<b>Appendix E. IBM-Supplied Programs</b> . . . . .	E-1
QEXCVTDV . . . . .	E-1
QEXRCDBK . . . . .	E-2
QEXRMVDE . . . . .	E-2
QEXSHRO . . . . .	E-4
<b>Bibliography</b> . . . . .	H-1
General AS/400-Related Books . . . . .	H-1
Programming Language and Utility Books . . . . .	H-2
Communications Books . . . . .	H-3
Programming and Debugging Material . . . . .	H-3
Migration Books . . . . .	H-4
<b>Index</b> . . . . .	X-1



---

## Notices

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

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

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

Address your questions to:

IBM Corporation  
Software Interoperability Coordinator  
3605 Highway 52 N  
Rochester, MN 55901-9986 USA

This publication could contain technical inaccuracies or typographical errors.

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

Changes or additions to the text are indicated by a vertical line (|) to the left of the change or addition.

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

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

---

## Trademarks And Service Marks

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

Advanced Function Printing	Operating System/2
AFP	Operating System/400
Application System/400	OS/2
APPN	OS/400
AS/400	PROFS
CICS	RPG/400
DisplayWrite	SAA
IBM	Systems Application Architecture
Intelligent Printer Data Stream	System/370
IPDS	System/36
NetView	VTAM
OfficeVision	400

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

---

## About System/36 Environment Reference (SC41-4731)

This book contains information for creating procedures for the AS/400 system in the OS/400 System/36 environment. It also provides reference information about how to identify and use procedures, control commands, operation control language (OCL) statements, and System/36 environment utility programs.

For information about other AS/400 publications, see either of the following:

- The *Publications Reference* book, SC41-4003, in the AS/400 Softcopy Library.
- The *AS/400 Information Directory*, a unique, multimedia interface to a searchable database containing descriptions of titles available from IBM or from selected other publishers. The *AS/400 Information Directory* is shipped with your system at no charge.

For a list of related publications, see the “Bibliography.”

---

## Who Should Use This Book

This book is intended primarily for the AS/400 system application programmers and operators who work with the AS/400 system in the OS/400 System/36 environment.

Before you use this book, you should be familiar with the information in the following publications:

- The book, *System Operation*, contains introductory material about the AS/400 system. You should read this book first if you are not familiar with the AS/400 system.
- The book, *System/36 Environment Programming*, describes how the OS/400 System/36 environment functions. It also contains information about techniques to use when programming for the AS/400 system in the System/36 environment.
- The book, *System/36 Migration Planning*, provides information for planning the details of migration and performing the functions of the Migration Utility.



---

# Chapter 1. Introduction

This book describes the procedures, control commands, operation control language (OCL) statements, and procedure control expressions supplied with the System/36 environment function of the IBM\* Operating System/400\* (OS/400\*) and other licensed programs.

---

## Procedures

A **procedure** is a collection of statements that cause one or more programs to run. You use procedures to do a task on the system, such as listing the contents of a disk file or running a program. Procedures make it possible to avoid entering frequently used statements each time they are required. A **command** is an instruction that tells the system to do something.

To run a procedure, enter a **procedure command** at a keyboard. A procedure command contains the name of the procedure to run and optional information defining the function the procedure performs. For example, enter the following to run an IBM-supplied procedure named SYSLIST:

```
SYSLIST
```

This procedure command contains only the name of the procedure to run.

You usually enter procedure commands with information that tells the procedure what to do. The following example shows a procedure command that runs an IBM-supplied procedure named LISTLIBR that lists a library member named PAYROLL:

```
LISTLIBR PAYROLL
```

Another way to run a procedure is to create a menu. Menus allow you to enter a number that corresponds to the procedure command, rather than having to enter the procedure command. See the book, *ADTS/400: Screen Design Aid for the System/36 Environment* for more information about creating menus.

Many procedures are supplied as part of the System/36 environment and the other licensed program products. The System/36 environment procedures allow you to do the following:

- Create data files and libraries.
- Save, restore, and copy data files and libraries.
- List information about data files, libraries, and the system.

The procedures allow you to create and change library members and to compile programs. See Chapter 4, "Procedures," for detailed information on the System/36 environment IBM-supplied procedures.

You can make your own procedures. See Chapter 2, "Making Your Own Procedures," for this information.

---

## OCL Statements

The **operation control language** (OCL) statements provide the System/36 environment with all the information it needs about job runs. OCL statements are normally contained in procedures, although you can enter them from a keyboard. See Chapter 5, "OCL Statements," for detailed information about OCL statements.

---

## System/36 Environment Utility Programs and Their Control Statements

IBM supplies utility programs as part of the System/36 environment to do certain functions, such as copying a disk file, listing a library member, or communicating with another system.

When a System/36 environment utility program runs, OCL statements identify the program and supply additional required information. Besides OCL statements, **utility control statements** define the functions the utility program does. Normally, a procedure contains the OCL statements and utility control statements required to do a job.

The following example shows the statements needed to list a library member named PAYROLL. The \$MAINT utility program runs. The LOAD and RUN statements are OCL statements. The COPY and END statements are utility control statements.

```
// LOAD $MAINT
// RUN
// COPY FROM-PAYLIB,TO-PRINT,NAME-PAYROLL,LIBRARY-S
// END
```

The statements in the example indicate the following:

<b>LOAD</b>	This OCL statement loads \$MAINT into main storage. \$MAINT is one of the System/36 environment utility programs.								
<b>RUN</b>	This OCL statement starts running the \$MAINT program.								
<b>COPY</b>	This utility control statement passes the following information to the \$MAINT program: <table><tr><td>FROM-PAYLIB</td><td>The library that contains the member is named PAYLIB.</td></tr><tr><td>TO-PRINT</td><td>The information prints.</td></tr><tr><td>NAME-PAYROLL</td><td>A library member named PAYROLL is used.</td></tr><tr><td>LIBRARY-S</td><td>The library member is a source member.</td></tr></table>	FROM-PAYLIB	The library that contains the member is named PAYLIB.	TO-PRINT	The information prints.	NAME-PAYROLL	A library member named PAYROLL is used.	LIBRARY-S	The library member is a source member.
FROM-PAYLIB	The library that contains the member is named PAYLIB.								
TO-PRINT	The information prints.								
NAME-PAYROLL	A library member named PAYROLL is used.								
LIBRARY-S	The library member is a source member.								
<b>END</b>	This utility control statement indicates that no more utility control statements follow.								

The utility programs you can use for each procedure are described in Appendix A, "System/36 Environment Utility Programs."

---

## Control Commands

Use control commands to control the system, the printers, and the display stations. The following example stops the printing of spooled output on printer P3:

```
STOP PRT,P3
```

See Chapter 6, “Control Commands,” for detailed descriptions of the control commands.

---

## Concept Information and Programming Considerations

For introductory information about computers for the Application System/400\* (AS/400\*) system, see the book, *System Operation*. For information about system concepts and programming techniques in the System/36 environment, see the *System/36 Environment Programming* book.

---

## Conventions Used for Describing Syntax Formats

When syntax formats are shown in this book, capitalized expressions, brackets, braces, parentheses, and underscoring have special meanings.

### Capitalized Expressions

Enter capitalized expressions as they are shown in the syntax formats. Also enter numbers and special characters within a capitalized expression as they are shown. Replace an expression that is not capitalized with a value that is appropriate. For example:

```
// FORMS LINES-value
```

59020001.0

could be coded:

```
// FORMS LINES-66
```

### Braces { }

Braces in a syntax format are not coded as part of the command or statement. Braces indicate that you must code one of the values enclosed within the braces. For example:

```
// DATE { mmddy  
         { ddmyy  
         { yymmdd }
```

59020002.0

indicates that if you choose to code a date, it must be in one of the three formats shown: mmddy, ddmyy, or yymmdd. For example, enter April 14, 1989 as 041489 (mmddy), 140489 (ddmyy), or 890414 (yymmdd).

## Brackets [ ]

Brackets in a syntax format are not coded as part of the command or statement. Brackets indicate that the expression they enclose is optional. If a list of values is enclosed in brackets, you can choose not to code a value or to code one of the items in the list. For example:

```
PASSWORD [CHANGE]
```



59020511-1

indicates you do not need to code the parameter, but if you choose to code the parameter, it must be the CHANGE format.

For example:

```
PASSWORD CHANGE
```

## Parentheses ( )

Parentheses in a syntax format are not coded as part of the command or statement. Parentheses indicate that the value enclosed within the parentheses is an abbreviation. You can enter the abbreviation in place of the characters above the parentheses. For example:

```
STATUS (D) SESSION (S)
```



59020004-0

indicates that you can enter STATUS SESSION as any of the following:

```
STATUS SESSION
```

```
D S
```

```
STATUS S
```

```
D SESSION
```

## Underscores \_

Underscoring identifies default values. The system automatically uses the default value if you do not code an optional value. For example:

```
OFF [DROP  
HOLD]
```



59020005-0

indicates that the system assumes DROP if you do not code the parameter. (Remember, the brackets indicate the parameter is optional.)

## Commas ,

The syntax formats often indicate that commas are required, when the commas are not shown inside brackets. The commas are shown outside the brackets to remind you that if you omit a parameter, you must enter a comma to indicate the position of the omitted parameter (when one or more parameters are coded in positions that follow the omitted parameter). For example:

```
PRINT  [ printer id ] , [ lines per page ] , [ lpi value ] , [ cpi value ] ,  
       [ forms number ] , [ degree of rotation ] , [ paper drawer ]
```

59020184-1

indicates that if you do not code the second parameter but do code the third parameter, you could code a comma indicating the place of the missing parameter. For example:

```
PRINT SYSTEM, ,15
```

For any procedure or control commands you enter, commas following the last parameter coded are optional. For example:

```
PRINT SYSTEM,,,,,
```

and

```
PRINT SYSTEM
```

are treated the same.

## Example Syntax Diagram

The following syntax diagram shows how you can combine diagramming methods:

```
// ALLOCATE UNIT- { T1  
                  T2 } [ ,AUTO- { YES } ] [ ,CONTINUE- { NO } ] [ ,WAIT- { NO } ]  
                  ' T1 , T2
```

59020634-1

To allocate unit T1 with CONTINUE as YES and WAIT as NO, enter:

```
// ALLOCATE UNIT-T1,AUTO-YES,CONTINUE-YES,WAIT-NO
```

Note that NO is the default for the WAIT parameter, so if you enter the following:

```
// ALLOCATE UNIT-T1,AUTO-YES,CONTINUE-YES
```

the results are the same.

---

## Directory to Using System Support

When you are signed on to the AS/400 system, you can use the system help support to help you do a task. The system help support includes menus, prompt displays, and help text. The menus allow you to select a task to perform. When you select an item from a menu, either:

- Another menu appears (as you select options, each one gets more specific about the task you perform).
- A prompt display for a procedure or command appears.

A prompt display allows you to enter the necessary parameters and then run a procedure or command to do the task. The help text explains the menus, the menu options, the procedures and commands, and the parameters for the procedures and commands.

## Creating and Maintaining Disk Files

### Changing

To change the information in a disk file using the data file utility (DFU), see “UPDATE Procedure” on page 4-294.

To use a file in a program, see “FILE OCL Statement (for Disk Files)” on page 5-25.

### Copying

To copy a disk file and to do one or more of the following, see “COPYDATA Procedure” on page 4-50:

- Create a new disk file with the same file organization.
- Create a new disk file with a different file organization.
- Change the record length of a file.
- Change the position and length of the keys in an indexed file.
- Include specific records in the new file.
- Omit specific records from the new file.
- Remove deleted records from a file.

To add a disk file to an existing file on diskette, see “SAVE Procedure” on page 4-220.

To create or copy a basic data exchange or I-exchange diskette file (to transfer a file to another system), see “TRANSFER Procedure” on page 4-289.

To copy a disk file to a tape file (to transfer a file to another system) or a tape file to a disk file, see “TAPECOPY Procedure” on page 4-255.

To copy a spool file(s) to a disk file, see “COPYPRT Procedure” on page 4-57.

### Creating

To create a disk file and enter records into the file using the data file utility (DFU), see “ENTER Procedure” on page 4-89.

To create or use a file in a program, see “FILE OCL Statement (for Disk Files)” on page 5-25.

To create an empty file, see “BLDFILE Procedure” on page 4-17. To create an alternative index file over a physical file, see “BLDINDEX Procedure” on page 4-20.

### **Organizing**

See “COPYDATA Procedure” on page 4-50.

### **Printing, Displaying**

To print or display the contents of a disk file, see “LISTDATA Procedure” on page 4-137 or “LISTFILE Procedure” on page 4-142.

To print or display the names of all the files on a disk, diskette, tape, or tape cartridge, see “CATALOG Procedure” on page 4-29.

To print or display the contents of a file saved on diskette, tape, or tape cartridge, see “LISTDATA Procedure” on page 4-137 or “LISTFILE Procedure” on page 4-142.

To print the contents of a file using the data file utility (DFU), see “LIST Procedure” on page 4-134. To display the contents of a file using DFU, see “INQUIRY Procedure” on page 4-121.

To print or display a spool file that was copied to a user file, see “COPYPRT Procedure” on page 4-57.

To print a graphics file on an intelligent printer data stream (IPDS) printer, see “PRTGRAPH Procedure” on page 4-186.

### **Removing**

To remove a file from disk or diskette, see “DELETE Procedure” on page 4-76.

To copy a disk file and remove deleted records from the file, see “COPYDATA Procedure” on page 4-50.

### **Renaming**

To rename a disk file, see “RENAME Procedure” on page 4-194.

### **Saving**

To save one or more disk files on diskette, tape, or tape cartridge, see “SAVE Procedure” on page 4-220.

To add a disk file to an existing file on diskette, see “SAVE Procedure” on page 4-220.

### **Restoring**

To restore one or more diskette, tape, or tape cartridge files to disk, see “RESTORE Procedure” on page 4-206.

### **Sorting**

To sort the contents of a disk file, see “SORT Procedure” on page 4-247.

To sort the index keys of an indexed file, see “KEYSORT Procedure” on page 4-129.

## **Creating and Maintaining Extended Character Files**

### **Changing**

To change the information in an extended character file, see “CGU Procedure” on page 4-39.

### **Creating**

To create double-byte characters for the extended character file, see “CGU Procedure” on page 4-39.

### **Sorting**

To sort the extended character file, see “SRTX Procedure” on page 4-248.

## **Creating and Maintaining Libraries**

### **Changing**

To change a library member using the source entry utility (SEU), see “SEU Procedure” on page 4-242.

To change the current library at your display station, see “SLIB Procedure” on page 4-244.

To change the sign-on library for your display station, see “SET Procedure” on page 4-235.

To change the name, subtype, or reference number of a library member, see “CHNGEMEM Procedure” on page 4-40.

### **Copying**

To copy one or more library members from one library to another library, see “LIBRLIBR Procedure” on page 4-129.

To copy one or more library members to a disk, diskette, tape, or tape cartridge file, see “FROMLIBR Procedure” on page 4-103.

To copy a library member to a basic data exchange diskette file to transfer the library member to another system, see “Copy Members from a Library (FROMLIBR Procedure)” on page A-45.

To copy a disk, diskette, tape, or tape cartridge file containing one or more library members to a library, see “TOLIBR Procedure” on page 4-283.

### **Creating**

To create a new procedure or source member in a library using the source entry utility (SEU), see “SEU Procedure” on page 4-242.

### **Printing, Displaying**

To print or display the names of the members in a library, or to list the contents of one or more library members, see “LISTLIBR Procedure” on page 4-147.

To print or display the names of all the libraries on a disk, diskette, tape, or tape cartridge, see “CATALOG Procedure” on page 4-29.

To print or display the names of files that contain saved library members or the names of library members saved to these files on disk, diskette, tape, or tape cartridge, see “LISTFILE Procedure” on page 4-142.

### **Removing**

To remove a library from disk or diskette, see “DELETE Procedure” on page 4-76.

To remove one or more members from a library, see “REMOVE Procedure” on page 4-192.

### **Renaming**

To rename a library, see “RENAME Procedure” on page 4-194.

To rename a library member, see “CHNGEMEM Procedure” on page 4-40.

**Restoring**

To restore a library from diskette, tape, or tape cartridge to disk, see “RESTLIBR Procedure” on page 4-203.

**Saving**

To save a library on diskette, tape, or tape cartridge, see “SAVELIBR Procedure” on page 4-228.

## Maintaining Folders and Folder Members

**Creating**

To create or maintain a folder, see “TEXTFLDR Procedure” on page 4-280.

To create or maintain a folder member, see “TEXTDOC Procedure” on page 4-264.

**Printing, Displaying**

To print or display the contents of a folder member, see “TEXTDOC Procedure” on page 4-264.

To print or display the names of all the folders on a disk, diskette, or tape cartridge, see “CATALOG Procedure” on page 4-29.

**Removing**

To remove a folder from disk, see “DELETE Procedure” on page 4-76.

**Renaming**

To rename a folder member, see “TEXTDOC Procedure” on page 4-264. To rename a folder, see “RENAME Procedure” on page 4-194.

## Processing Diskettes

**Allocating**

To allocate the diskette drive to a job, see the “ALLOCATE OCL Statement” on page 5-6.

If you do not want to allocate the diskette drive, see “DEALLOC OCL Statement” on page 5-21.

**Copying**

To copy all of a diskette to another diskette, see “COPYI1 Procedure” on page 4-55.

To create or copy a basic data exchange or I-format diskette file, see “TRANSFER Procedure” on page 4-289.

To copy a diskette file containing one or more library members to a library, see “TOLIBR Procedure” on page 4-283.

To copy one or more library members from a library to diskette, see “FROMLIBR Procedure” on page 4-103.

**Preparing**

To prepare (also called *initialize*) a diskette before using it to save data, see “INIT Procedure” on page 4-119.

**Printing, Displaying**

To list the names of files, libraries, and folders contained on a diskette, or to list general information about a diskette, see “CATALOG Procedure” on page 4-29.

To list the contents of a file saved on a diskette, see “LISTDATA Procedure” on page 4-137 or “LISTFILE Procedure” on page 4-142.

To list a library saved on a diskette or to list a diskette exchange file, see “LISTFILE Procedure” on page 4-142.

### **Removing**

To remove one or more files or libraries from diskette, see “DELETE Procedure” on page 4-76.

### **Restoring**

To restore one or more files saved on diskette back to disk, see “RESTORE Procedure” on page 4-206.

To restore a library from diskette to disk, see “RESTLIBR Procedure” on page 4-203.

### **Saving**

To save one or more disk files on diskette, see “SAVE Procedure” on page 4-220.

To save a library on diskette, see “SAVELIBR Procedure” on page 4-228.

## **Processing Tapes**

### **Allocating**

To allocate the tape drive to a job, see “ALLOCATE OCL Statement” on page 5-6.

To release the tape drive, see “DEALLOC OCL Statement” on page 5-21.

### **Copying**

To create or copy a tape exchange file, see “TAPECOPY Procedure” on page 4-255.

To copy a tape file containing one or more library members to a library, see “TOLIBR Procedure” on page 4-283. To copy one or more library members from a library to tape, see “FROMLIBR Procedure” on page 4-103.

### **Preparing**

To prepare (also called *initialize*) a tape or tape cartridge before using it to save data, see “TAPEINIT Procedure” on page 4-262.

### **Printing, Displaying**

To list the names of files, libraries, and folders contained on a tape or tape cartridge, or to list general information about a tape or tape cartridge, see “CATALOG Procedure” on page 4-29.

To list the contents of a file saved on a tape or tape cartridge, see “LISTDATA Procedure” on page 4-137 or “LISTFILE Procedure” on page 4-142.

To list a library saved on a tape or tape cartridge or to list a tape exchange file, see “LISTFILE Procedure” on page 4-142.

### **Restoring**

To restore one or more files saved on tape or tape cartridge back to disk, see “RESTORE Procedure” on page 4-206.

To restore a library from tape or tape cartridge to disk, see “RESTLIBR Procedure” on page 4-203.

**Saving**

To save one or more disk files on tape or tape cartridge, see “SAVE Procedure” on page 4-220.

To save a library on tape or tape cartridge, see “SAVELIBR Procedure” on page 4-228.

## **Creating and Maintaining Display Formats, Menus, and Message Members**

**Display Formats**

To generate only display formats from source statements, see “FORMAT Procedure” on page 4-99.

To remove a display format from a display format load member, see “FORMAT Procedure” on page 4-99.

**Menus**

To generate a menu from source statements, see “BLDMENU Procedure” on page 4-25.

To display a menu, see “MENU Control Command” on page 6-14.

**Message Members**

To generate a message member from source statements, see “CREATE Procedure” on page 4-66.

To create or change the automatic response values for system or user message members, see “RESPONSE Procedure” on page 4-196.

To assign a message member to a procedure or a program, see the “MEMBER OCL Statement” on page 5-59.

## **Creating and Maintaining Programs**

**RPG II**

To compile an RPG II source program, see “RPGC Procedure” on page 4-214.

To compile an RPG II source program that contains auto report specifications, see “AUTOC Procedure” on page 4-11.

To create or change a procedure or an RPG II or auto report source member, see “RPGSEU Procedure” on page 4-219.

To create a cross reference listing of an RPG II program, see “RPGX Procedure” on page 4-219.

To create or change display formats for an RPG program, see “RPGSDA Procedure” on page 4-218.

To create display formats for an RPG II CONSOLE file, see “RPGR Procedure” on page 4-217.

**COBOL**

To compile a COBOL source program, see “COBOLC Procedure” on page 4-44.

To create or change display formats for a COBOL program, see “COBSDA Procedure” on page 4-48.

To create or change a procedure or COBOL source member, see “COBSEU Procedure” on page 4-48.

## Using the Office Products

### Office Products

To change the default values used by the office products procedures, see “OFCDFLT Procedure” on page 4-165.

To create or maintain a calendar, see “OFCCAL Procedure” on page 4-163.

To enroll or change the enrollment of the office product users, see “OFCUSER Procedure” on page 4-171.

To file an online document or log the receipt of a printed document, see “OFCFILE Procedure” on page 4-166.

To maintain library descriptions used for libraries, see “OFCLDF Procedure” on page 4-167.

To select different ways of looking at the directory, see “OFCDIR Procedure” on page 4-165.

To send messages to a group, see “OFCMSG Procedure” on page 4-169.

To view a list of the library requests submitted, see “OFCSTAT Procedure” on page 4-171.

To work with mail, see “OFCMAIL Procedure” on page 4-167.

To work with a user group, see “OFCGRP Procedure” on page 4-166.

To add or update data on a file, see “OFCSRCH Procedure” on page 4-170.

### Query

To add or update data on a file, see “QRYDE Procedure” on page 4-187.

To define a query or work with already existing queries, see “QRY Procedure” on page 4-187.

To run a query and display, print, or send to disk the data produced, see “QRYRUN Procedure” on page 4-188.

### Word Processing

To create or maintain a folder, see “TEXTFLDR Procedure” on page 4-280.

To create or maintain a folder member, see the “TEXTDOC Procedure” on page 4-264.

To create or maintain text profiles, see “TEXTPROF Procedure” on page 4-281.

To maintain a document object in a folder, see “TEXTOBJ Procedure” on page 4-281.

To maintain a supplemental dictionary, see “TEXTDCT Procedure” on page 4-264.

To perform various print tasks, see “TEXTPRTQ Procedure” on page 4-281.

To release documents held for later printing, see “TEXTPRTQ Procedure” on page 4-281.

## Using the Personal Computer

To send or receive messages by way of a personal computer, see “MSG OCL Statement” on page 5-62 or “MSG Control Command” on page 6-15.

To start the PC utility, see “PCU Procedure” on page 4-177.

## Defining Data in Files

### Interactive Data Definition Utility (IDDU)

To create or maintain a data dictionary, see “IDDUUCT Procedure” on page 4-116.

To create or maintain a data definition, see “IDDUDFN Procedure” on page 4-116.

To create or enter data in a disk file, see “IDDUUDISK Procedure” on page 4-116.

To link or unlink a file definition with a disk file, see “IDDUULINK Procedure” on page 4-116.

To print field, format, or file definitions in a data dictionary, see “IDDUUPRT Procedure” on page 4-117.

## Running Programs and Procedures

### Date

To change the program, job, or job step date, see “DATE OCL Statement” on page 5-19 or “DATE Procedure” on page 4-71.

### Data Files

To use a data file in a program, see “FILE OCL Statement (for Disk Files)” on page 5-25.

### Display Formats

To show a display format from a procedure to prompt for parameters or data, see “PROMPT OCL Statement” on page 5-77.

### Display Stations

To assign a display station to a program, see “WORKSTN OCL Statement” on page 5-105.

### Local Data Area

To change information in the local data area, see “LOCAL OCL Statement” on page 5-56.

To substitute information from the local data area into a procedure, see “?L'position,length'? (Local Data Area) Expression” on page 3-19.

### Menus

To display and use a menu, see “MENU Control Command” on page 6-14 or “MENU OCL Statement” on page 5-61.

### Messages

To assign a message member to a program or procedure, see “MEMBER OCL Statement” on page 5-59.

To send a message to an operator, see:

“MSG Control Command” on page 6-15

“// \* (Informational Message) Statement” on page 3-44

“PAUSE Statement” on page 3-53  
“ERR Procedure” on page 4-93  
“// \*\* (System Console Message) Statement” on page 3-45  
“MSG OCL Statement” on page 5-62

To specify whether informational messages display, see “INFOMSG OCL Statement” on page 5-51.

To change the automatic response level for a procedure or program, see “NOHALT OCL Statement” on page 5-65.

### **Printers**

To use a printer in a program, see “PRINTER OCL Statement” on page 5-68.

To change one or more of the following, see “PRINT Procedure” on page 4-178, “SET Procedure” on page 4-235, or “FORMS OCL Statement” on page 5-46:

- The printer used
- The number of lines per page
- The number of characters per inch (cpi) printed horizontally
- The number of lines per inch (LPI) printed vertically
- The forms number used
- The orientation or size of the output on the page
- The printer drawer from which to use paper

To change the current system list device, see “SYSLIST Procedure” on page 4-253 or “SYSLIST OCL Statement” on page 5-98.

To specify whether a border or heading is to be printed with Print key output, see “PRINTKEY Procedure” on page 4-182 or the “WORKSTN OCL Statement” on page 5-105.

### **Priority**

To change the processing priority of a job, see “ATTR OCL Statement” on page 5-8.

### **Starting**

To display the current status of running programs and procedures, use the STATUS USERS control command.

To load and run a compiled program, see “LOAD OCL Statement” on page 5-55 and “RUN OCL Statement” on page 5-83.

To start a procedure, see:

- “INCLUDE OCL Statement” on page 5-49
- “JOBQ Control Command” on page 6-12
- “JOBQ OCL Statement” on page 5-52
- “EVOKE OCL Statement” on page 5-23

To start a job from a job stream, see “JOBSTR Procedure” on page 4-125.

### **Stopping**

To stop a currently running job, see “CANCEL Control Command” on page 6-3.

### **Switches**

Switches are also called external indicators U1 through U8.

To change the switch settings, see “SWITCH (Switches) Condition” on page 3-39.

To check the switch settings in a procedure, see “Substitution Expressions” on page 3-7.

### **Waiting**

To wait a specific amount of time or to wait until a certain time occurs before a job begins, see “WAIT OCL Statement” on page 5-103.

## **Changing and Controlling Printers, Jobs, and Display Stations**

To display information about your display station environment, see “STATUS Control Command” on page 6-24.

### **Displayed Data**

To display the current system list device, use the STATUS SESSION control command.

To change the current system list device, see “SYSLIST Procedure” on page 4-253 or “SYSLIST OCL Statement” on page 5-98.

### **Job Queue**

To display the status of the job queue, use the STATUS JOBQ control command.

To place a job on the job queue, see “JOBQ Control Command” on page 6-12 or “JOBQ OCL Statement” on page 5-52.

To change the number of jobs allowed to run from the job queue, see “CHANGE Control Command” on page 6-5.

To stop the job queue or a specific job queue priority, see “STOP Control Command” on page 6-32.

To start the job queue or a specific job queue priority, see “START Control Command” on page 6-22.

### **Jobs**

To display the status of the programs and procedures running on the system, use the STATUS USERS control command.

To start a job, see:

“INCLUDE OCL Statement” on page 5-49

“JOBQ Control Command” on page 6-12

“JOBQ OCL Statement” on page 5-52

“EVOKE OCL Statement” on page 5-23

To end a job, see “CANCEL Control Command” on page 6-3.

### **Libraries**

To display your current or session library, use the STATUS SESSION control command.

To change the current or session library, see “SLIB Procedure” on page 4-244 or “LIBRARY OCL Statement” on page 5-54.

### **Menus**

To display a user menu, see “MENU Control Command” on page 6-14 or “MENU OCL Statement” on page 5-61.

To display a help menu, see “HELP Procedure” on page 4-109.

## Messages

To specify whether informational messages display, see “INFOMSG Control Command” on page 6-12.

To change the automatic response level for your display station, see “NOHALT Procedure” on page 4-161.

To send a message to another display station, or to display a message sent from another display station, see “MSG Control Command” on page 6-15.

## Printed Data

To display the status of printed output on the spool file, use the STATUS PRT control command.

To display the status of the spool writers, use the STATUS WRT control command.

To change one or more of the following, see “PRINT Procedure” on page 4-178, “SET Procedure” on page 4-235, or “FORMS OCL Statement” on page 5-46:

- The printer used
- The number of lines per page
- The number of characters per inch (cpi) printed horizontally
- The number of lines per inch (LPI) printed vertically
- The forms number used
- The orientation or size of the output on the page
- The printer drawer from which paper is used

To change the current system list device, see “SYSLIST Procedure” on page 4-253.

To change your sign on or current printer, see “SET Procedure” on page 4-235.

To specify whether a border or heading is to be printed with Print key output, see the “PRINTKEY Procedure” on page 4-182 or the “WORKSTN OCL Statement” on page 5-105.

To start the printing of spooled output, see “START Control Command” on page 6-22 or “START OCL Statement” on page 5-95.

To stop the printing of spooled output, see “STOP Control Command” on page 6-32 or “STOP OCL Statement” on page 5-96.

To restart the printing of spooled output, see “RESTART Control Command” on page 6-20.

To hold spooled output on the spool file to prevent it from printing, see “HOLD Control Command” on page 6-11.

To release held spooled output for printing, see “RELEASE Control Command” on page 6-19.

To change one or more of the following, see “CHANGE Control Command” on page 6-5 or “CHANGE OCL Statement” on page 5-14:

- The position of spool file entries
- The number of copies printed
- The forms number used
- The printer to be used
- The number of separator pages printed

Whether spool file entries should print before they are closed

To cancel one or more entries from the spool file, see “CANCEL Control Command” on page 6-3 or “CANCEL OCL Statement” on page 5-13.

### **Signing Off**

To sign your display station off the system, see “OFF Control Command” on page 6-18 or “OFF OCL Statement” on page 5-66.

## **Starting Procedures Automatically**

### **Starting Up**

To run procedures automatically, immediately after starting up the system, see “#STRUP1 Procedure” on page 4-3 and “#STRUP2 Procedure” on page 4-3.

## **Communicating with Other Systems**

### **Remote Job Entry**

To do remote job entry, see “MSRJE Procedure” on page 4-159.

To print the information created by a remote job entry task, see “RJFILE Procedure” on page 4-213.

To define a remote job entry control table, see “RJTABLE Procedure” on page 4-213.

To sign an IBM personal computer on to SNA 3270 device emulation, see “EP3270 Procedure” on page 4-92.

There are many ways to communicate with a remote system using your own communications applications, including the following:

- Advanced program-to-program communications (APPC)
- SNA upline facility (SNUF)
- Asynchronous communications
- BSC equivalence link (BSCCL)
- Retail communications
- Finance communications

For more information see “SESSION OCL Statement” on page 5-84 in this book and the communications chapter in the *System/36 Environment Programming* book.

## **Maintaining the System**

### **Disk Space**

To list the contents of the disk and the amount of unused space on the disk, see “CATALOG Procedure” on page 4-29.

## **Determining and Correcting Problems**

### **Debugging**

To cause each operation control language (OCL) statement processed in a procedure to be logged to the job log, see “LOG OCL Statement” on page 5-58.

To cause all evaluations of the procedure control expressions to be listed as the procedure is run, see “DEBUG OCL Statement” on page 5-22.

---

## Reference Numbers for Library Members

The **reference number** is a 6-digit number that exists in the directory entry for all procedure and source library members in the System/36 environment.

On the AS/400 system this reference number can be changed with the following procedures or commands:

- CHNGEMEM procedure (\$MAINT utility)
- Change System/36 Procedure Attributes (CHGS36PRCA) command
- Change System/36 Source Attributes (CHGS36SRCA) command
- Edit System/36 Procedure Attributes (EDTS36PRCA) command
- Edit System/36 Source Attributes (EDTS36SRCA) command
- Work with System/36 Procedure Attributes (WRKS36PRCA) command
- Work with System/36 Source Attributes (WRKS36SRCA) command

The reference number is initially set to 000000 when a new procedure or source member is created. The reference number is copied along with the contents of a member when using any of the System/36 environment functions (FROMLIBR, TOLIBR, LISTLIBR, and so on) or the AS/400 Save System/36 Library Members (SAVS36LIBM) and Restore System/36 Library Members (RSTS36LIBM) commands. When a procedure or source member is migrated from System/36, the reference number is copied to the AS/400 system along with the contents of the member.

The reference number is shown on the listing produced by the LISTLIBR procedure when the directory or the contents of procedure or source members are listed.

The reference number is not updated by the Start Source Entry Utility (STRSEU) command when a procedure or source member is edited.

The reference number is placed in the first 6 characters of the object description of the load member produced when the display format source specifications are compiled into a display format load member using the FORMAT procedure or the Create System/36 Display File (CRTS36DSPF) command, and of the message file created by the CREATE procedure or the Create System/36 Message File (CRTS36MSGF) command.

**Note:** The reference number is not placed in the load member produced when a source member is compiled into a program (RPG II and COBOL).

---

## Chapter 2. Making Your Own Procedures

This chapter describes how to write and use your own procedures.

---

### What a Procedure Is

A **procedure** is a collection of statements that causes one or more programs to run. Procedures make it possible to avoid entering several statements each time you must perform a job. The collection of statements is stored in a library member called a procedure member.

The purpose of a procedure is to do a job. The procedure can do this job by having one or more job steps. A **job step** is a unit of work done by one program. A job step usually begins with the LOAD OCL statement, and usually ends with the RUN OCL statement (some OS/400 utility program job steps end with an END utility control statement). The following procedure contains one job step because only one program loads and runs:

```
// LOAD PROG1  
// RUN
```

The following example has two job steps because two programs load and run:

```
// LOAD PROG1  
// RUN  
// LOAD PROG2  
// RUN
```

---

### What a Procedure Can Contain

The statements in a procedure control the files, display stations, and printers used by a program. For example:

```
// LOAD PROG3  
// FILE NAME-CUSTOMER  
// RUN
```

These statements indicate the following:

**LOAD** The program to be run is named PROG3.

**FILE** PROG3 uses a disk file named CUSTOMER.

**RUN** The program runs.

Procedures are made up of the following types of statements:

- **Operational control language (OCL) statements**, which you use to load and run programs. OCL statements also indicate how the System/36 environment runs the program and how the System/36 environment uses the input and output devices that the program may require. See Chapter 5, “OCL Statements,” for more information about the OCL statements used in this chapter. Examples of OCL statements are LOAD, FILE, and RUN.
- **Procedure control expressions**, which control how the procedure processes based upon certain conditions. See Chapter 3, “Procedure Control Expressions,” for more information about the procedure control expressions used in this chapter.
- **Procedure commands**, which cause other procedures to run. Procedures supplied by IBM as part of the System/36 environment and as part of the other licensed programs are described in Chapter 4, “Procedures.” Examples of these procedures are COPYDATA and LISTLIBR. You can also use your own procedures.
- **Utility control statements** for System/36 environment utility programs, which pass information to utility programs. The utility control statements are shown in Appendix A, “System/36 Environment Utility Programs.”
- **Control language (CL) commands**, which perform OS/400 functions. See “Control Language Commands in Procedures” on page 2-20.

Procedures cannot contain any control commands. See Chapter 6, “Control Commands” for descriptions of the control commands.

---

## Entering Procedures into the System

Enter procedures into a library using the source entry utility (SEU). SEU is described in detail in the *ADTS/400: Source Entry Utility* book.

You can also use the \$MAINT utility program to enter procedures into a library. Procedures are stored as members of a source file named QS36PRC. See “\$MAINT Utility” on page A-37 for information.

---

## Naming Procedures

A procedure name can consist of any combination of 1 to 8 characters. The first character must be alphabetic (A through Z, #, \$, or @). The remaining characters must follow OS/400 object-naming restrictions.

Any procedure names you specify in the System/36 environment that have special characters are quoted internally.

When procedure names are processed in the System/36 environment, the syntax of the procedure name is checked. If the procedure name fails the syntax check, an invalid name error message is sent. If the procedure name meets the System/36 naming conventions, the System/36 environment determines whether the name should be quoted.

If the procedure name is a simple name, it is not quoted internally. Simple names follow these naming conventions:

- The first character is a capital A through Z, \$, #, or @.
- The remaining characters are capital A through Z, \$, #, @, digits 0 through 9, underscore, or period.

If the procedure name is an extended name, it is quoted internally. Extended names can contain any character that can be displayed (code point greater than hex 3F) except embedded blanks, asterisk (\*), single quote ('), double quote ("), or question mark (?).

If the procedure name contains embedded blanks, asterisk (\*), single quote ('), double quote ("), or question mark (?), the procedure name is invalid and the System/36 environment sends an error message.

Remember to enter a library parameter in the procedure statement if a user-written procedure has the same name as a system procedure. Otherwise, the system automatically runs the procedure that exists in the current user library or library list. Refer to "INCLUDE OCL Statement" on page 5-49 for more information on the library parameter for a procedure statement.

When you use the HELP procedure to specify parameters for a system procedure, and a user procedure with the same name exists in the current user library or library list, the system procedure Help display appears, but the user procedure runs. This results in the user procedure using the system procedure parameters. Unless the parameters match, the user procedure fails to run properly.

---

## Procedure Parameters

You can define parameters for your procedures. These parameters allow information and variables to pass to the procedure. Parameters can have up to 128 characters.

A procedure can have a maximum of 64 parameters. **Positional parameters** are parameters passed to procedures. Whenever a parameter appears in a procedure command, it must appear in the same position in relation to other parameters in the procedure command. Each parameter is assigned a place, such as the first parameter or the second parameter. If you omit a parameter, you must still use a comma to indicate the *position* of the omitted parameter.

To use parameters in your own procedures, use substitution expressions. Use the following substitution expressions for procedure parameters:

Expression	Meaning
?n?	This expression substitutes the value of the nth positional parameter into the statement in the procedure. For example, the value entered on the procedure command in the first parameter position is substituted by the expression ?1?.
?n'value'?	If you do not enter the nth parameter (you do not specify it on the procedure command that starts the procedure), this expression defines the value of the nth parameter and substitutes the value into the statement in the procedure. Once you set a parameter to

'value' it remains at that value as if you had entered it. It is not a temporary substitution.

If you enter the nth parameter, this expression substitutes the value of the nth parameter into the statement in the procedure.

?nT'value'? If you do not enter the nth parameter, this expression temporarily substitutes the value specified into the statement in the procedure. The temporary value is used only for the current substitution.

If you enter the nth parameter, this expression substitutes the value of the nth parameter into the statement in the procedure.

?nF'value'? This expression changes the value of the nth parameter and substitutes that new value into the statement in the procedure. Once a parameter is set to 'value', it remains at that value as if you had entered it. It is not a temporary substitution.

?nR? If you do not enter the nth parameter, this expression displays a message to enter the required parameter and allows the operator to enter it. For parameters for which you receive a prompt from the system, you can enter a maximum of 60 characters. The system then gives the nth positional parameter a value and substitutes that value into the statement in the procedure.

If the nth parameter has a value, no message displays, and the value of the nth positional parameter substitutes into the statement in the procedure.

?nR'mic'? If you do not enter the nth parameter, this expression displays a message you can specify and allows the operator to enter the parameter. For parameters for which you receive a prompt from the system, you can enter a maximum of 60 characters. The system then defines the nth positional parameter and substitutes the value you entered into the statement in the procedure. See "CREATE Procedure" on page 4-66 for information about how to create these messages.

If you enter the nth parameter, no message displays and the value of the nth positional parameter substitutes into the statement in the procedure.

The resolved values are placed in the statement starting at the position of the left-most question mark (?).

### Example

This example assumes that you created a program that reads and prints the information contained in a file. You can specify the name of the file as a parameter.

Create a procedure named PROC1 to run your program (the program is named PRTFILE). Use SEU, DSU, or the \$MAINT utility to enter this procedure into the system. Parameter 1 is the name of the disk file to use. Create the procedure PROC1 as follows:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-?1?
// RUN
```

Enter the PROC1 procedure into a library procedure member named PROC1. The statements in the PROC1 procedure indicate the following:

- LOAD** The program to run is named PRTFILE.
- FILE** The program uses a disk file (UNIT-F1). The program refers to the file as INPUT. Parameter 1 contains the name of the file (indicated by LABEL-?1?).
- RUN** The program starts running.

When you enter the following command to run the PROC1 procedure:

```
PROC1 FILEA
```

the following statements run:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-FILEA
// RUN
```

Note that FILEA was substituted in place of the ?1? expression.

## Procedure Parameter Defaults

You can define defaults for parameters. A **default** is a value that automatically substitutes for an omitted or undefined parameter. Indicate defaults for parameters using the ?n'value'? substitution expression.

### Example

This example assumes that you want to change PROC1 so that the default for the file name is EMPLOYES if you do not enter a parameter. Create a procedure named PROC2 (based upon PROC1) as follows:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-?1'EMPLOYES'?
// RUN
```

Enter the PROC2 procedure into a library procedure member named PROC2 using SEU or DSU. The statements in the PROC2 procedure indicate the following:

- LOAD** Same as in PROC1.
- FILE** The program refers to the disk file as INPUT. The name of the file to be used is contained in parameter 1. If you did not enter a value for the first parameter when you started the procedure, a value of EMPLOYES is assumed for the LABEL parameter. That is, EMPLOYES is the default for the first parameter (indicated by LABEL-?1'EMPLOYES'?).
- RUN** Same as in PROC1.

When you enter the following command to run the PROC2 procedure:

```
PROC2
```

the following statements run:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-EMPLOYES
// RUN
```

Note that because you did not specify a value for the first parameter, the value EMPLOYES was assumed.

But when you enter the following command to run the PROC2 procedure:

```
PROC2 FILEA
```

the following statements run:

```
// LOAD PRTFILE  
// FILE NAME-INPUT,UNIT-F1,LABEL-FILEA  
// RUN
```

Note that because you specified a value for the first parameter, that value (FILEA) was used.

## Testing Entered Parameters

Use the IF conditional expressions to check the parameters entered for a procedure. This lets you ensure that the parameters are correct before you run your program.

The IF expression has two forms:

- IF or IFT test for true conditions
- IFF tests for false conditions

### Example

The following procedure requires that the first parameter (the name of the file to process) be either EMPLOY or CUSTOM. If the first parameter is not one of these values, only the PAUSE and CANCEL statements process (which causes an error message to display and the procedure to cancel).

```
// IF ?1?=EMPLOY GOTO OK  
// IF ?1?=CUSTOM GOTO OK  
// PAUSE 'Parameter 1 must be EMPLOY or CUSTOM'  
// CANCEL  
// TAG OK  
//     LOAD     PRTFILE  
//     FILE     NAME-INPUT,UNIT-F1,LABEL-?1?  
//     PRINTER  NAME-OUTPUT,DEVICE-?2'P3'  
//     RUN
```

## Parameter Coding Considerations

The blank, comma (,), apostrophe ('), question mark (?), slash (/), equal sign (=), plus sign (+), greater than sign (>), and hyphen (-) have special meanings in procedures, OCL statements, and utility control statements. You can use these characters in parameters for a procedure, but only with caution.

---

## Continuing the Lines of a Procedure

Enter a maximum line length of 120 characters from the keyboard for a library procedure member or statement. If you need to use a statement of more than 120 characters, use the continuation symbol to continue the statement on one or more lines. The continuation symbol is a plus sign (+) in the position where the incomplete line ends. Use the plus sign to continue statements in procedures and to continue lines you enter at the keyboard. The plus sign works as a continuation character when no characters, other than blanks, appear to the right of the plus sign.

The continuation symbol causes the next line (starting with the first nonblank character) to concatenate (append) to the previous line. Continue as many lines as you wish, but you cannot exceed 512 characters in the resulting concatenated statement.

For example, a procedure contains the following three lines:

```
// IFF ?1?=EMPLOY +  
    IFF ?1?=MANAG +  
    PROCA TEST,RUN
```

This results in the following statement:

```
// IFF ?1?=EMPLOY IFF ?1?=MANAG PROCA TEST,RUN
```

Note that the blanks after the + sign and the blanks before each continued line are ignored. The blanks before the + sign are preserved. Also notice that no comment data was placed after the + signs. Had there been comments on these lines, the + sign would have been ignored. You can place comment data on the last line only because it is not continued.

You can specify the continuation symbol anywhere within the line, such as within a parameter, substitution expression, or any other type of data. For example, a procedure could contain the following two lines:

```
// IFF ?1?=EMPLOY I+  
    FF ?1?=MANAG PROCA TEST,RUN
```

The resulting statement would be:

```
// IFF ?1?=EMPLOY IFF ?1?=MANAG PROCA TEST,RUN
```

The plus sign (+) provides continuation in addition to the continuation allowed in OCL statements and utility control expressions.

You can use the continuation expression on all procedures, OCL statements, and utility control statements. However, you cannot use it on a comment (\*) statement. You can specify the + sign anywhere within the line; for example, within a parameter, substitution expression, or in the middle of a word. However, to make your procedures easier to read and maintain, continue your statements only at the end of words and expressions.

If a record ends with a shift-in character just before the continuation expression, and the first nonblank character of the next record is a shift-out character, both the shift-in and shift-out characters are removed.

---

## Calling a Procedure from Another Procedure

One procedure can call another procedure. A procedure called by another procedure is a **nested procedure**. This is generally helpful when a job calls the same procedure several times within the job. You could enter and store the procedure only once and then call it as often as necessary.

Suppose, for example, that a procedure named PAYROLL contains (in addition to other statements) a TAXES procedure command, and that another procedure named TAXES contains FEDER and STATE procedure commands. The system calls and runs the TAXES, FEDER, and STATE procedures when the operator enters the PAYROLL procedure command.

The four procedures are as follows:

---

<b>PAYROLL Procedure</b>	<b>TAXES Procedure</b>	<b>FEDER Procedure</b>	<b>STATE Procedure</b>
* PAYROLL	* TAXES	* FEDER	* STATE
// ...	// ...	// ...	// ...
// ...	// ...	// ...	// ...
TAXES	FEDER		
// ...	// ...		
// ...	// ...		
	STATE		
	// ...		
	// ...		

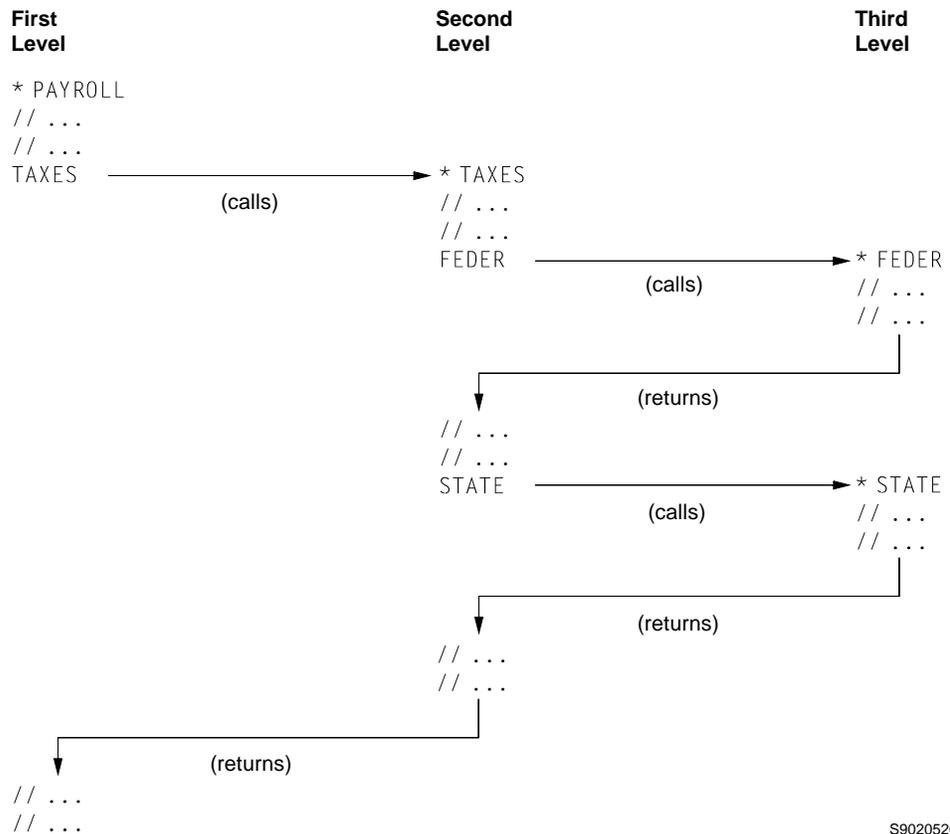
---

In this example, TAXES, FEDER, and STATE are nested procedures. When a nested procedure ends, processing returns to the next statement in the calling procedure.

This example contains three levels of procedures: the first level contains PAYROLL, the second level contains TAXES, and the third level contains FEDER and STATE. One level can contain more than one procedure command, but no more than 255 levels of procedures are allowed. The procedures are called as shown in the following figure:

Operator  
Enters

PAYROLL



S9020520-0

## Procedure Attributes

When a procedure is created by the \$MAINT utility program, or changed when using the Edit System/36 Procedure Attributes (EDTS36PRCA) or Change System/36 Procedure Attributes (CHGS36PRCA) commands, you can assign attributes to specify the following:

- Whether the procedure is a multiple requester terminal (MRT) procedure. For information about MRT programs and procedures, see the *System/36 Environment Programming* book. Normally, procedures are single requester terminal procedures.
- If the procedure is a MRT procedure that is not a never-ending program (also called a non-NEP MRT), whether termination of the MRT should be delayed after the last device is released from the MRT, just in case a new device is about to route to the MRT. The delay value for terminating non-NEP MRT procedures can be displayed by using the Display System/36 (DSPS36) command, or changed by using the Change System/36 (CHGS36) command.
- Whether OCL statements should be logged to the job log. See “LOG Procedure” on page 4-155 to prevent the procedure’s OCL statements from logging onto the job log when the procedure runs. The procedure command that started a procedure always logs to the job log. Normally, the procedure

command and the OCL statements for your procedures are all logged to the job log. This is done to help you debug your procedures.

- Whether to pass parameters to the procedure or to pass data to the program. If the program data attribute is set, the data on the procedure command passes to the first program run by the procedure. The data passes on the first display station input or read request. The data starts with the first nonblank character following the procedure or library name and ends with the last nonblank character in the statement. For example:

```
PAYROLL DATA01
```

The PAYROLL procedure could have 2 data fields:

- Positions 1 through 4 (which contains DATA)
- Positions 5 and 6 (which contains 01)

Every MRT procedure has this attribute even though the attribute may not have been selected when the procedure was created. Normally, single requester terminal procedures pass parameters to the procedure.

- The logical record length of the statements in the procedure member (used by Save System/36 Library Member [SAVS36LIBM] command).
- The reference number assigned to the procedure member.

---

## Example Procedures

This section provides examples to help you write your own procedures.

### Example 1: Procedure SAMPLE

This example shows a procedure named SAMPLE. This is a somewhat simple procedure because it has only two parameters: the file used and the creation date of the file. No licensed program supplies this procedure, and you must enter it if you want to test it. The procedure runs the \$COPY utility program, and has a function similar to the LISTDATA procedure. For more information about the statements in this procedure, see “Listing \$COPY Files (LISTDATA/LISTFILE Procedures)” on page A-11.

```
SAMPLE file name, [mmddyy  
                  ddmmyy  
                  yymmdd]
```

59020008-0

**file name** This specifies the name of a file to list. This is a required parameter. If you omit it, the procedure sends you a prompt.

**mmddyy, ddmmyy, or yymmdd**

This specifies the creation date of the file.

The statements in the SAMPLE procedure are as follows:

```
* SAMPLE Procedure
* Parameter 1: File name (required, prompted for if omitted)
* Parameter 2: File's creation date (optional)
// * 'SAMPLE PROCEDURE RUNNING' 1
// LOAD $COPY 2
// FILE NAME-COPYIN,LABEL-?1R?, 3
// IFF ??= DATE-??, 4
// UNIT-F1 5
// RUN 6
// COPYFILE OUTPUT-CHAR 7
// END 8
```

### Description of the Lines in the SAMPLE Procedure

The lines contained in the SAMPLE procedure are described as follows. For more information about the OCL statements in this procedure, see Chapter 5, “OCL Statements.” For more information about the COPYFILE utility control statement in this procedure, see “Listing \$COPY Files (LISTDATA/LISTFILE Procedures)” on page A-11.

Line	Description
1	Displays a message indicating the procedure is running.
2	Loads the \$COPY utility program into main storage.
3	Defines the name of the file listed. If you did not enter the first parameter, you receive a prompt using the ?1R? expression.
4	If you specified a date, the DATE parameter is included.
5	Specifies the file as a disk file. Since the UNIT parameter does not end with a comma, it is the last parameter of the FILE statement. This allows the DATE parameter to be optional. That is, if you did not specify a date parameter, the DATE parameter does not generate.
6	Starts running the \$COPY program, which then reads the COPYFILE and END utility control statements.
7	Causes the printable characters in the file to list.
8	Indicates the end of the utility control statements to the \$COPY program.

### Example 2: Procedure LISTKEYS

This example shows a procedure named LISTKEYS. It is a somewhat more difficult procedure because it contains six parameters, two of which have defaults. No licensed program supplies this procedure, and you must enter the procedure if you want to run it. The procedure runs the \$COPY utility program and has a function similar to the LISTDATA procedure. For more information about the statements in this procedure, see “Listing \$COPY Files (LISTDATA/LISTFILE Procedures)” on page A-11.

```
LISTKEYS file name, [mmddyymmddyy], [KEY PKY], [first key], [last key], [CHAR HEX CRT]
```

S9020635-0

**file name** This specifies the name of an indexed file that lists. You must specify this parameter. If you omit it, you receive a prompt for it from the procedure.

**mmddy, ddmmy, or yymmdd**

This specifies the creation date of the file.

**KEY** This specifies that the indexed file has normal (unpacked) keys. If you do not specify a parameter, KEY is assumed.

**PKY** This specifies that the indexed file has packed keys.

**first key** This specifies the first key to list in the file. If you do not specify a parameter, the file lists, starting with the first key in the file.

**last key** This specifies the last key to list in the file. If you do not specify a parameter, the file lists, ending with the last key in the file. You can only specify a last key if you specify a first key.

**CHAR** This specifies a listing of the printable characters in the file. Any unprintable characters list as blanks. If you do not specify a parameter, CHAR is assumed.

**HEX** This specifies listing of both the printable characters and their hexadecimal representations. Any unprintable characters list as blanks, but their hexadecimal representations list.

**CRT** This specifies that the records display.

The statements in the LISTKEYS procedure are as follows:

```
* LISTKEYS Procedure
// MEMBER USER1-MESSAGES,LIBRARY-MYLIB 1
// IF JOBQ-NO IF EVOKED-NO * 0001 2
// IF ?1?= EVALUATE ?1R'0002'? 3
// IF ?1?= RESET LISTKEYS ,?2?,?3?,?4?,?5?,?6? 4
// EVALUATE ?3'KEY'? ?6'CHAR'? 5
// IFF ?3?=KEY IFF ?3?=PKY ERR 0003,3 6
// IFF ?6?=CHAR IFF ?6?=HEX +
// IFF ?6?=CRT ERR 0004,3 7
// IF ?4?= IFF ?5?= ERR 0005,3 8
// LOAD $COPY 9
// FILE NAME-COPYIN,LABEL-?1?, 10
// IFF ?2?= DATE-?2?, 11
// UNIT-F1 12
// RUN 13
// COPYFILE OUTPUT-?6?,REORG-YES 14
// IFF ?4?= IF ?5?= SELECT ?3?,FROM-?4? 15
// IFF ?4?= IFF ?5?= SELECT ?3?,FROM-?4?,TO-?5? 16
// END 17
// RETURN 18
```

\*

\* Parameters:

- \* 1 File name (required, prompted for if omitted)
- \* 2 File's creation date (optional)
- \* 3 KEY or PKY. Indicates normal or packed keys (default: KEY)
- \* 4 First key to list (default: first key in file)
- \* 5 Last key to list (default: last key in file)
- \* 6 CHAR, HEX, or CRT. Indicates how the file is to be listed. (default: CHAR)

The LISTKEYS procedure uses five messages from a message member named MESSAGES, which is in a library named MYLIB. The messages in MESSAGES are as follows:

```
MESSAGES,1
0001 The LISTKEYS procedure is running
0002 Enter the name of the indexed file to list:
0003 The third parameter must be KEY, PKY, or blank
0004 The sixth parameter must be CHAR, HEX, CRT, or blank
0005 A last key is only allowed if a first key is entered
```

For more information about message members, see “CREATE Procedure” on page 4-66.

### Description of the Lines in the LISTKEYS Procedure

The following describes the lines in the LISTKEYS procedure. For more information about the OCL statements in this procedure, see Chapter 5, “OCL Statements.” For more information about the COPYFILE and SELECT utility control statements in this procedure, see “Listing \$COPY Files (LISTDATA/LISTFILE Procedures)” on page A-11.

Line	Description
<b>1</b>	Indicates the first-level user message member (named MESSAGES).
<b>2</b>	Displays message 0001 if you enter the procedure from the keyboard.
<b>3</b>	If you omit the file name, message 0002 displays and asks for the file name.
<b>4</b>	If the operator does not enter a name, the procedure restarts. Any parameters the operator entered are included.
<b>5</b>	If you do not specify a third parameter, KEY is assumed. If you do not specify a sixth parameter, CHAR is assumed.
<b>6</b>	If the third parameter is not KEY or PKY, error message 0003 displays. The message has an error option of 3. Accept the 3 option and cancel the procedure.
<b>7</b>	If the sixth parameter is not CHAR, HEX, or CRT, an error message 0004 displays. Note how the plus sign (+) is used to continue this statement onto two lines. The message has an error option of 3. Accept the 3 option and cancel the procedure.
<b>8</b>	If you do not specify a fourth parameter but do specify a fifth parameter, error message 0005 displays. The message has an error option of 3. Accept the 3 option and cancel the procedure.
<b>9</b>	Loads the \$COPY utility program into main storage.
<b>10</b>	Defines the name of the file to list.
<b>11</b>	If you specify a date, the DATE parameter is included.
<b>12</b>	Specifies the file as a disk file. The UNIT parameter does not end with a comma; it is the last parameter of the FILE statement. This allows the DATE parameter to be optional. That is, if you did not specify a date parameter, the DATE parameter does not generate.
<b>13</b>	Starts running the \$COPY program, which then reads the COPYFILE, SELECT, and END utility control statements.

- 14** Specifies how the file lists (OUTPUT-?6?). REORG-YES specifies listing of the records sequentially by index key.
- 15** If you enter parameter 4 and do not enter parameter 5, the file lists starting with the specified key. If you do not enter parameters 4 and 5, the entire file lists.
- 16** If you enter both parameters 4 and 5, the file lists starting with the specified key and ending with the specified key.
- 17** Indicates the end of the utility control statements to the \$COPY program.
- 18** Ends the procedure. The system does not read the comment statements following the RETURN.

### Example 3: Procedure SCRNPRT

This example shows a procedure named SCRNPRT. This procedure calls a program that reads a file and prints information from the file. The procedure has five parameters and shows a prompt display if the operator does not enter any parameters. The procedure also checks each parameter to ensure that its value is correct. If the procedure finds an incorrect parameter, the prompt display appears with a message indicating the parameter in error. Also, the cursor positions to that parameter and the parameter is highlighted.

No licensed program supplies this procedure. You must enter it if you want to run it.

```
SCRNPRT file name, [ printer id ], [ copies ], [ 10 ], [ 6 ]
                  [ P1 ], [ 1 ], [ 15 ], [ 8 ]
                  [ P2 ]
```

S9020010-0

If you do not enter a first parameter, a prompt display appears.

**file name** This specifies the name of a file that lists.

**printer id** This specifies the printer ID to use. Specify either P1 or P2. P1 is the default.

**copies** This specifies the number of copies to print. Enter any number from 1 to 5. 1 is the default.

**10 or 15** This specifies the characters per inch value to use. 10 is the default.

**6 or 8** This specifies the lines per inch value to use. 6 is the default.

Figure 2-1 shows a listing of the SCRNPRT procedure.

```
*
// EVALUATE ?2'P1'? ?3'1'? ?4'10'? ?5'6'? 2
*
// IFF ?1? = GOTO CHKFILE 3
*
// IF ?L'1,8'? = SCRNSPOL EVALUATE P1 = ?L'10,8'? Default for parm 1 4
// IF ?L'1,7'? = SCRNPRT EVALUATE P1 = ?L'10,8'? Default for parm 1
*
// TAG PROMPT 5
*
// PROMPT MEMBER-SCRN,FORMAT-SCRNPRT,LENGTH-'8,2,1,1,75'
// IF ?CD? = 2007 RETURN Cmd? - End procedure
*
* Reset error indicators
// EVALUATE P31 = P32 = P33 = P34 = P35 = 6
*
// IFF ?1? = GOTO CHKFILE 7
// EVALUATE P6 = 'You must enter a file name.' P31 = 'X'
// GOTO PROMPT
*
// TAG CHKFILE 8
*
// IF DATAF1-?1? GOTO FILEOK
// EVALUATE P6 = 'File ?1? does not exist, enter a different name.' P31 = 'X'
// GOTO PROMPT
*
// TAG FILEOK 9
// LOCAL OFFSET-10,BLANK-8,DATA-'?1?'
*
// IFF ?2? = P1 IFF ?2? = P2 IFF ?2? = PX GOTO P2ERR 10
// GOTO P3CHK
// TAG P2ERR
// EVALUATE P6 = 'Printer ID must be P1 or P2, re-enter value.' P32 = 'X'
// GOTO PROMPT
*
// TAG P3CHK 11
*
// IFF ?3? = 1 IFF ?3? = 2 IFF ?3? = 3 IFF ?3? = 4 ?3? = 5 GOTO P3ERR
// GOTO P4CHK
// TAG P3ERR
// EVALUATE P6 = 'Number of copies must be 1 through 5,' P33 = 'X'
// GOTO PROMPT
*
// TAG P4CHK 12
*
// IFF ?4? = 10 IFF ?4? = 15 GOTO P4ERR
// GOTO P5CHK
// TAG P4ERR
// EVALUATE P6 = 'Character per inch value must be 10 or 15.' P34 = 'X'
// GOTO PROMPT
```

Figure 2-1 (Part 1 of 2). Listing of SCRNPRT Procedure

```

*
// TAG P5CHK 13
*
// IFF ?5?=6 IFF ?5?=8 GOTO P5ERR
// GOTO RUN
// TAG P5ERR
// EVALUATE P6='Line per inch value must be 6 or 8.' P35='X'
// GOTO PROMPT
*
// TAG RUN 14
*
// LOCAL BLANK-*ALL,DATA-'SCRNPRT'           Procedure name
// LOCAL OFFSET-10,DATA-'?1?'               File name
// PRINTER NAME-SCRNPRT,DEVICE-???, 15
//          COPIES-???,CPI-?4?,LPI-?5?
// IF ?5?=8 LINES-88
// ELSE LINES-66
*
// INFOMSG NO 16
*
// LOAD SCRNPRT,TEMLIB 17
// FILE NAME-SCRNPRTF,UNIT-F1,LABEL-?1?
// RUN
*
// INFOMSG YES 18
// * 'Displays printed.'

```

Figure 2-1 (Part 2 of 2). Listing of SCRNPRT Procedure

## Description of the SCRNPRT Procedure

The lines called out in the SCRNPRT procedure in Figure 2-1 on page 2-15 are described as follows. For more information about the OCL statements in this procedure, see Chapter 5, “OCL Statements.”

Line Group	Description
<b>1</b>	Comment information. Describes the procedure in general and describes each parameter.
<b>2</b>	Sets the defaults for parameters 2 through 5.
<b>3</b>	If you enter parameter 1 (the file name), the GOTO statement bypasses the initial showing of the prompt display.
<b>4</b>	If the operator did not enter a name, the file name stored in the local data area is assumed. The name is stored in positions 10 through 17 by either the SCRNPRT procedure or by another procedure called SCRNSPOL.
<b>5</b>	Shows the prompt display. The display format load member is named SCRNPRT. The format is named SCRNPRT (to match the procedure name).  If the operator presses command key 7, the procedure ends.
<b>6</b>	Clears the parameters used for the display format error indicators.

- 7** If you do not enter the file name on the prompt display, parameter 6 sets to a message and the prompt display appears. Parameter 31 is given a value that sets on indicator 31, positions the cursor at the file name field, and shows the field in reverse image.
- 8** If you enter a nonexistent file name, parameter 6 sets to a message and the prompt display appears. Parameter 31 is given a value that sets on indicator 31, positions the cursor at the file name field, and shows the field in reverse image.
- 9** If the file exists, the name is stored in the local data area.
- 10** If you do not enter a printer ID of P1 or P2, parameter 6 sets to a message and the prompt display appears. Parameter 32 is given a value that sets on indicator 32, positions the cursor at the printer ID field, and shows the field in reverse image.
- 11** If you do not enter a number of copies value of 1 through 5, parameter 6 sets to a message and the prompt display appears. Parameter 33 is given a value, which sets on indicator 33, positions the cursor at the number of copies field, and shows the field in reverse image.
- 12** If you do not enter a characters per inch value of 10 or 15, parameter 6 sets to a message and the prompt display appears. Parameter 34 is given a value that sets on indicator 34, positions the cursor at the characters per inch field, and shows the field in reverse image.
- 13** If you do not enter a lines per inch value of 6 or 8, parameter 6 sets to a message and the prompt display appears. Parameter 35 is given a value that sets on indicator 35, positions the cursor at the lines per inch field, and shows the field in reverse image.
- 14** Stores the procedure name and file name in the local data area.
- 15** Controls the output for a printer file (SCRNPRT) created by the RPG program executed at **17**.
- 16** Causes informational messages not to be displayed.
- 17** Runs an RPG program that reads the file and prints information stored in the file.
- 18** Causes an informational message to be displayed again.

The following is an example of the SCRNPRT display:

```
SCRNPRT PROGRAM

This program lists the Print key display images contained in a disk file.

Name of the disk file that contains the display images . . . . . _____
ID of printer to be used . . . . . P1,P2 P1
Number of copies to print . . . . . 1-5 1
Character per inch value (15 valid for P1 only) . . . . . 10,15 10
Line per inch value (8 valid for P1 only) . . . . . 6,8 6

Cmd7 - End program
```

The S- and D-specifications for the display format shown by the procedure are shown in the following figure. For more information about display formats, see the *System Operation* book.



---

## Control Language Commands in Procedures

Control language (CL) commands can be included in System/36 procedures. A CL command can be used in the System/36 environment if it is valid in the environment in which it is running (batch or interactive). Commands that are allowed only in a CL program (such as DCL, MONMSG, or RTVJOB), and commands that are valid only in a batch job stream (such as BCHJOB, DATA, or ENDBCHJOB) cannot be processed in the System/36 environment.

CL commands are valid anywhere an OCL statement or procedure command is valid. They are not valid where a utility control statement or inline source statement is required (after a // RUN and before a // END or /\*).

The System/36 environment first checks each statement to determine if it is a valid System/36 OCL statement or command. If not, it is processed as a CL command. If you wish to process a CL command that has the same name as a System/36 procedure or command, you must enter a command label or a library qualifier on the command to prevent it from being processed as a System/36 statement. Assume that you have a System/36 procedure and a CL command that are named CMD1. The following examples show how a statement using this name is processed.

<b>CMD1 [parameters]</b>	Process the System/36 procedure.
<b>CMD1,USERLIB [parameters]</b>	Process the System/36 procedure from library USERLIB.
<b>A:CMD1 [parameters]</b>	Process the CL command.
<b>*LIBL/CMD1 [parameters]</b>	Process the CL command.
<b>USERLIB/CMD1 [parameters]</b>	Process the CL command from library USERLIB.

## Using Substitution Expressions on CL Commands

System/36 substitution expressions can be used on any part of a CL command, including the command name and keyword names. Substitution is performed before the statement type is determined. You must ensure that the syntax for a CL command will be valid after the substitution is done, as in the following example:

DSP?1? ?1?(?2?) ?3?

The following table shows the resulting commands for various values of the positional parameters:

<b>?1?</b>	<b>?2?</b>	<b>?3?</b>	<b>Resulting Command</b>
CMD	TEST		DSPCMD CMD(TEST)
PGM	LIB1/PGMA	OUTPUT(*PRINT)	DSPPGM PGM(LIB1/PGMA) OUTPUT(*PRINT)
LIB	#LIBRARY		DSPLIB LIB(#LIBRARY)

Four substitution expressions were created for use with CL commands. These substitution expressions are valid only for the System/36 environment, and are not valid on System/36. The expressions are as follows:

- Files library (?FLIB?)
- Session files library (?SFLIB?)
- Message ID (?MSGID?)
- Device name (?DEV'UNIT'?)

These substitution expressions are described in the following sections.

### **Files Library (?FLIB?) Expression**

The ?FLIB? expression is replaced with the name of the current files library (up to 10 characters) that contains the database files used by the System/36 environment. ?FLIB? can be used to ensure that CL commands and programs that use database files use the same files as those used by System/36 environment functions, as shown in the following example:

```
// LOAD S36PGM
// FILE NAME-MASTER, LABEL-TEMP1
// RUN
OVRDBF FILE(MASTER) TOFILE(?FLIB?/TEMP1)
CALL OS400PGM
```

See “FILELIB OCL Statement” on page 5-43 and “FLIB Procedure” on page 4-98 for more information about the current files library.

### **Session Files Library (?SFLIB?) Expression**

The ?SFLIB? expression is replaced with the name of the session files library (up to 10 characters). The session files library is the files library that will be in use when the current System/36 job (outermost procedure) ends.

### **Message ID (?MSGID?) Expression**

The ?MSGID? expression is replaced with the 7-character message ID of the error message sent by the CL command processed in the System/36 environment. If the last CL command ended without an error message, or if no CL command has been processed, this expression is replaced with a null value. The following example shows how this expression can be used with the Check Object (CHKOBJ) command to determine whether a library exists:

```
CHKOBJ OBJ(TESTLIB) OBJTYPE(*LIB)
// IF ?MSGID?/CPF9801 CRTLIB LIB(TESTLIB)
```

### **Device Name (?DEV'unit'?) Expression**

The ?DEV'unit'? expression is replaced with the 10-character name of the device equivalent to the 2-character System/36 environment unit ID. If the specified unit ID is not configured, the device name expression is replaced with a null value. The following example shows how you can use this expression to substitute the name of the work station device from which the job runs:

```
SNDSMSG MSG('JOB COMPLETED SUCCESSFULLY') TOMSGQ(?DEV'?WS'??)
```

The following example shows how the ALCOBJ command can be used to allocate the device known to the System/36 environment as T1:

```
ALCOBJ OBJ((?DEV'T1'? *DEVD *DEVD))
```

## Prompting for CL Commands

Prompt for CL commands in a procedure by entering a question mark before the command name or by entering one of the selective prompt characters (??, ?\*, or ?-) before keywords. A prompted CL command runs only in an interactive job. See the *CL Reference* book for more information about prompt characters.

## Conditioning CL Commands

CL commands are valid as the statement on IF and ELSE procedure control expressions. For example:

```
// IF EVOKED-NO IF JOBQ-NO DSPPFM FILE(?1?) MBR(???)  
// ELSE CPYF FROMFILE(?1?) TOFILE(*PRINT) MBR(???)
```

Unlike OCL statements, it is not possible to condition parts of a CL command by placing each part on a separate line with a comma continuation character. If a CL command requires more than one line in the procedure, continue it with a plus sign (+). The minus sign (-) is not a valid continuation character for commands in a procedure.

## Special Characters in Names

The System/36 environment allows special characters in names. The OS/400 program requires names containing special characters be enclosed in quotation marks (""). When you specify a name with special characters on a System/36 statement, do not specify the quotation marks. When you specify a name on a CL command that contains special characters, you must specify the quotation marks. If a name specified on a CL command is enclosed in quotation marks but does not contain any special characters, the quotation marks are ignored. For example, if "ABC" is specified, the OS/400 program will use the value ABC for the name. See the *CL Reference* book for the OS/400 naming rules.

If your installation uses special characters in names, put quotation marks around substitution expressions used as a name on a CL command. Doing this ensures the name is valid if the substitution expression contains special characters. In the following example, an error occurs if the file's library name or the parameter number 1 contains special characters:

```
CLRPFM FILE(?FLIB?/F?1?) MBR(*LAST)
```

If you change the foregoing statement to the following, the statement works correctly if either value contains special characters:

```
CLRPFM FILE('?FLIB?'/ 'F?1?') MBR(*LAST)
```

## Handling Errors on CL Commands

**Warning:** The default error handling for CL commands in a procedure is to issue a halt. The default error handling can be changed with the CHGS36 command. If you change the default to continue with the next statement, and an error on a CL command in a procedure prevents the procedure from continuing, error handling must be added (CHGS36MSGL command or ?MSGID?).

The CHGS36MSGL command can be used to determine what action to take when a CL command ends with an error message. Specific actions may be specified for certain error messages, as well as a default action for any error message. The following actions can be specified:

<b>*CONTINUE</b>	Put the error message ID in the ?MSGID? substitution expression and continue with the next statement.
<b>*IGNORE</b>	Set the ?MSGID? substitution expression to null and continue with the next statement.
<b>*CANCEL</b>	Cancel the procedure. For interactive jobs, return control to the keyboard for the next statement.
<b>*GOTO</b>	Go to a // TAG statement with a specified label.
<b>*HALT</b>	Display a halt message to which the operator must reply. The allowable options may also be specified.

You can specify a list of message IDs and the action to be taken for each. You can also specify a default action to be taken if the message ID is not on the list or if no list is specified. Each procedure can have a default action; if no default action is specified for a procedure, the default action for the session is initially set to the value in the System/36 environment configuration. The DSPS36 command can be used to display this value, and the CHGS36 command can be used to change the value.

The following statement sets the default action for the session to issue a halt with options 0, 1, and 3 allowed. This statement can be typed in, selected as a menu option, or contained in a procedure. This is in effect until you either sign off or enter the ENDS36 command.

```
CHGS36MSGL DFTACN(*HALT 013) SCOPE(*SESSION)
```

The following statement sets the default action for the current System/36 job to cancel the job if an error occurs on a CL command. This statement must be in a procedure and remains in effect for the duration of the System/36 job. After the first-level procedure ends, the default action specified for the session is in effect.

```
CHGS36MSGL DFTACN(*CANCEL) SCOPE(*JOB)
```

The next example shows how a message list can be used in a procedure. The first CHGS36MSGL command sets a message list that will go to label NOTEXIST if message CPF9801 occurs, and to label NOTAUT if message CPF9802 or CPF9820 occurs. If any other exception occurs, a halt is issued with only option 3 allowed. The second CHGS36MSGL command replaces the message list. If message CPF2105 occurs, it is ignored. If any other error occurs, the ?MSGID? substitution expression is set. The line after the DLTF command tests the ?MSGID? substitution expression and handles the error that occurred. The third CHGS36MSGL command removes the message list. If any error occurs on the CALL command following it, the default action for the job is taken.

```

CHGS36MSGL MSGL((CPF9801 *GOTO NOTEXIST) +
                ((CPF9802 CPF9820) *GOTO NOTAUT) +
                (*ANY *HALT 3))
CHKOBJ ?2?/?1? *PGM
// GOTO OK
// TAG NOTEXIST
   (code to handle object does not exist messages)
// GOTO OK
// TAG NOTAUT
   (code to handle not authorized to object messages)
// TAG OK
CHGS36MSGL MSG((CPF2105 *IGNORE) (*ANY *CONTINUE))
DLTF ?FLIB?/?3?
// IFF ?MSGID?/ ... (handle error)
CHGS36MSGL MSGL(*NONE)
CALL PGMA

```

The CHGS36MSGL command can set the message list or default action for the previous procedure level. This allows you to write a procedure that can be started from other procedures in an application to set a common message list or default action. For example, if the following statement is placed in a procedure, the message list and default action specified on the command will be in effect for any procedure that starts it:

```

CHGS36MSGL ((CPF0001 *CANCEL) (CPF9801 *CONTINUE)) +
            DFTACN(*HALT 013) SCOPE(*PRVPRC)

```

---

## Procedure Performance Tips and Coding Techniques

After you know how to create procedures, use this section to improve the performance of your procedures.

- Use GOTO and TAG statements rather than several redundant IF expressions. Use one IF expression and a GOTO expression to reduce the time needed to evaluate several IF expressions. The statements skipped by the GOTO and TAG expressions are not processed. For example, rather than doing the following:

```

// IF ?1?/Y LOAD $MAINT
// IF ?1?/Y RUN
// IF ?1?/Y COPY FROM-#LIBRARY,NAME-TEST,LIBRARY-P,TO-PRINT
// IF ?1?/Y END
:

```

do the following, which avoids duplicating the tests for parameter 1 by using GOTO and TAG statements:

```

// IFF ?1?/Y GOTO A
//   LOAD $MAINT
//   RUN
//   COPY FROM-#LIBRARY,NAME-TEST,LIBRARY-P,TO-PRINT
//   END
// TAG A
:

```

- Use ELSE statements if you have more than one IF expression and only one of the expressions can be true. All ELSE statements are skipped after a true IF expression. For example, rather than doing the following, which processes all three statements even though only one of the statements is true:

```
// IF ??/T SWITCH 1XXXXXX
// IF ??/J SWITCH X1XXXXXX
// IF ??/S SWITCH XX1XXXXX
```

do the following, which stops processing after the first true condition:

```
// IF ??/T          SWITCH 1XXXXXX
// ELSE IF ??/J     SWITCH X1XXXXXX
// ELSE IF ??/S     SWITCH XX1XXXXX
```

- Combine IF expressions where possible. The remainder of a statement is not processed after a false condition. For example, rather than doing the following:

```
// IF ??/T GOTO NEXT
// IF ??/J GOTO NEXT
// IF ??/S GOTO NEXT
// GOTO ERROR
// TAG NEXT
:
// RETURN
// TAG ERROR
// PAUSE 'ERROR IN PARAMETER 2'
// CANCEL
```

Do the following, which checks the value of parameter 2 and, if it does not equal T, J, or S, the ERROR processes:

```
// IFF ??/T IFF ??/J IFF ??/S GOTO ERROR
:
// RETURN
// TAG ERROR
// PAUSE 'ERROR IN PARAMETER 2'
// CANCEL
```

- Avoid using the informational message (`// *`) statement to display prompting messages (such as: ENTER MEMBER NAME or ENTER LIBRARY NAME). Use the PROMPT OCL statement and a display format instead. The advantages are:
    - You can display more information.
    - You require fewer disk operations.
    - For remote display stations, fewer data transmissions are made. When the system processes an `// *` statement, it saves the current display contents, shows the message, and shows the display again after the procedure ends. When the system processes the PROMPT statement, it simply shows the display format without having to save the current display contents.
- See “PROMPT OCL Statement” on page 5-77 for more information.
- After you test your procedures, stop logging the OCL statements to the job log. You may only need to have the OCL statements logged when you create and test your procedure. Stop the logging in one of the following ways:
    - Use the Edit System/36 Procedure Attributes (EDTS36PRCA) or Change System/36 Procedure Attributes (CHGS36PRCA) command.
    - The LOG procedure or OCL statement can specify whether the statements should be logged. See “LOG Procedure” on page 4-155 or “LOG OCL Statement” on page 5-58.
  - Remove the comments within a procedure to improve the processing time of your System/36 procedures. Reading comments uses system time.

Keep available a copy of the fully commented procedure but use the procedure without comments for production.

- Use your own libraries for your applications. Run procedures and programs from a library other than the system library (#LIBRARY). The system library has a very large directory; therefore, the System/36 environment needs more time to search for a library member in the system library than for the same member in one of your libraries.

Also, the System/36 environment always searches the current library first. If it does not find the member, then it searches the system library and then the library list. See the *CL Reference* book for information about the library list. See “SLIB Procedure” on page 4-244 and “LIBRARY OCL Statement” on page 5-54 for more information about setting the current library.

- Use substitution expressions to concatenate values. For example:

```
// IFF ?1?/0 IFF ?1?/1 GOTO ERROR
// SWITCH XXX?1?XX?1?X
```

If the first parameter is 1, the SWITCH statement is:

```
// SWITCH XXX1XX1X
```

If the first parameter is 0, the SWITCH statement is:

```
// SWITCH XXX0XX0X
```

- Concatenate values to create unique names. For example, you can use the ?WS? expression, which substitutes the current display station ID, to create a file name that is unique for each display station:

```
// FILE NAME-OUTPUT,LABEL-FILEA?WS?
```

This allows more than one operator to use the procedure containing this statement because each display station has its own unique work file. The program refers to the output file as OUTPUT, and if an operator at display station W1 ran the procedure, the actual name of the file would be FILEAW1.

- Use IF conditional expressions to avoid making the system operator respond to an informational message when a procedure is sent to the job queue or when the procedure is started by the EVOKE OCL statement or an intersystem communications function (OS/400-ICF) evoke operation. OS/400-ICF will hereafter be referred to as ICF. For example:

```
// IF JOBQ-N0 IF EVOKED-N0 * 'Procedure running'
```

This example displays the message only when the procedure is run from the display station; that is, not from the job queue and not evoked.

- Change the value of a parameter. This allows an operator to use fewer key-strokes. For example:

```
// * 'ENTER 1 TO PROCESS MONTHLY; 2 TO PROCESS WEEKLY'
// IF      ?1R?=1 EVALUATE P1='MONTHLY'
// ELSE IF ?1?=2  EVALUATE P1='WEEKLY'
//      ELSE CANCEL
INVENTORY ?1?
```

If the operator enters 1, the procedure INVENTORY MONTHLY runs. If the operator enters 2, the procedure INVENTORY WEEKLY runs. If the operator enters neither 1 nor 2, the procedure cancels.

---

## Debugging Your Procedures

Use the DEBUG and LOG OCL statements to debug your procedures. See “DEBUG OCL Statement” on page 5-22 and “LOG OCL Statement” on page 5-58 for more information.



---

## Chapter 3. Procedure Control Expressions

This chapter describes the expressions and statements you use to control the processing of your procedures.

Place these expressions and statements anywhere among the OCL and utility control statements in a procedure. Procedure control expressions include the following:

- Comment statement
- Nested substitution expressions
- IF conditional expressions
- ELSE expressions
- Informational message statement
- System console message statement
- CANCEL statement
- EVALUATE statement
- GOTO and TAG statements
- PAUSE statement
- RESET statement
- RETURN statement

---

### Things You Can Do Using Procedure Control Expressions

This section lists several tasks you can do using procedure control expressions. The tasks are listed in the following groups:

- “Substituting Values and Information”
- “Displaying Messages or Display Formats” on page 3-3
- “Data File Information” on page 3-4
- “Library Information” on page 3-4
- “Diskette Information” on page 3-5
- “Tape Information” on page 3-5
- “Comparing and Evaluating Values and Branching in Procedures” on page 3-5
- “Testing the Procedure or Job Environment” on page 3-6
- “Ending Procedures” on page 3-6

### Substituting Values and Information

#### Command Keys

To check whether you pressed a command or function key, see “?CD? (Return Code) Expression” on page 3-13.

#### Date and Time

To substitute the current date, see “?DATE? (Program Date) Expression” on page 3-16.

To change the date, see “DATE OCL Statement” on page 5-19.

To substitute the current time, see “?TIME? (System Time) Expression” on page 3-23.

#### Library Names

To substitute the current library name, see “?CLIB? (Current Library) Expression” on page 3-16.

To substitute the name of the library currently used by the System/36 environment to hold data files, see “?FLIB? (Files Library) Expression” on page 3-19.

To substitute the name of the session files library, see “?SFLIB? (Session Files Library) Expression” on page 3-22.

To substitute the session library name, see “?SLIB? (Session Library) Expression” on page 3-22.

To change the session or current files library, see “FLIB Procedure” on page 4-98 or “FILELIB OCL Statement” on page 5-43. To change the session or the current library, see “SLIB Procedure” on page 4-244 or “LIBRARY OCL Statement” on page 5-54.

### **Local Data Area**

To substitute data from the local data area, see “?L'position,length'? (Local Data Area) Expression” on page 3-19.

To place information into the local data area, see “LOCAL OCL Statement” on page 5-56.

### **Message ID**

To substitute the message ID of the error message sent by an OS/400 command processed in the System/36 environment, see “?MSGID? (Message ID) Expression” on page 3-21.

### **Message Text**

To substitute data from a message member, see “?Mmic? or ?M'mic,position,length'? (Message Member) Expression” on page 3-20.

To specify which message member is used, see “MEMBER OCL Statement” on page 5-59.

### **Parameters**

To substitute the value of a parameter entered on a procedure statement, see the following:

- “?n? (Parameter) Expression” on page 3-8
- “?n'value'? (Default Parameter) Expression” on page 3-9
- “?nT'value'? (Temporary Value Parameter) Expression” on page 3-9
- “?nF'value'? (Forced Value Parameter) Expression” on page 3-10
- “?R? (Required Parameter) Expression” on page 3-10
- “?nR? (Missing Parameter) Expression” on page 3-11
- “?nR'mic'? (Missing Parameter Message) Expression” on page 3-12
- “EVALUATE Statement” on page 3-46

To substitute the length of a parameter, see “?Cn? (Parameter Length) Expression” on page 3-12.

### **Printers**

To substitute the printer ID of the session printer, see “?PRINTER? (Session Printer) Expression” on page 3-21.

To substitute the system list device (either a printer ID or whether the output prints or displays), see “?SYSLIST? (System List Device) Expression” on page 3-22.

To substitute the device name of a printer device, see “?DEV'unit'? (Device Name) Expression” on page 3-17.

To specify a printer, see “PRINT Procedure” on page 4-178 or “PRINTER OCL Statement” on page 5-68.

#### **Procedure or Menu**

To substitute the outermost (first-level) procedure name, see “?PROC? (First Level Procedure) Expression” on page 3-22.

To substitute the current user menu name, see “?MENU? (Current Menu) Expression” on page 3-21.

To cause a menu to display, see “MENU OCL Statement” on page 5-61.

#### **Return Codes**

To check whether a compilation was successful, see “?CD? (Return Code) Expression” on page 3-13.

#### **User ID**

To substitute the operator's user ID who runs the procedure, see “?USER? (Operator's User ID) Expression” on page 3-23.

#### **Work Station ID**

To substitute the work station ID that runs the procedure, see “?WS? (Display Station ID) Expression” on page 3-24.

To substitute the device name of a work station, see “?DEV'unit'? (Device Name) Expression” on page 3-17.

## **Displaying Messages or Display Formats**

#### **Display Formats**

To cause a display format to appear at the display station that runs the procedure, see “PROMPT OCL Statement” on page 5-77.

To determine which command key was pressed on a PROMPT display, see “?CD? (Return Code) Expression” on page 3-13.

#### **Messages**

To display an informational message at the display station that runs the procedure, see “// \* (Informational Message) Statement” on page 3-44.

To display a message at the system console, see “// \*\* (System Console Message) Statement” on page 3-45.

To display a message because you did not specify a required parameter, see “?nR? (Missing Parameter) Expression” on page 3-11, “?R? (Required Parameter) Expression” on page 3-10, or “?nR'mic'? (Missing Parameter Message) Expression” on page 3-12.

To display a message, and to pause for the operator to respond to the message, see “ERR Procedure” on page 4-93, or “PAUSE Statement” on page 3-53.

To specify which message member is used, see “MEMBER OCL Statement” on page 5-59. For additional information on dual language message members, see “\$MGBLD Utility” on page A-55.

## Data File Information

### File Existence

To determine whether a data file exists on the system, see “DATAF1 (Files and Libraries on Disk) Condition” on page 3-28.

To determine whether a data file exists on diskette, see “DATAI1 (Files on Diskette) Condition” on page 3-29.

To determine whether a data file exists on tape or tape cartridge, see “DATAT (Files on Tape) Condition” on page 3-30.

### File Size

To determine the allocated size of a data file, see “?F'S,name'? or ?F'S,name,date'? (File Size) Expression” on page 3-17.

To determine the actual number of records contained in a data file, see “?F'A,name'? or ?F'A,name,date'? (Actual File Size) Expression” on page 3-18.

### Space Available

To determine the number of disk blocks available, see “BLOCKS (Available Disk Space) Condition” on page 3-27.

## Library Information

### Library Names

To substitute the name of the current library, see “?CLIB? (Current Library) Expression” on page 3-16.

To substitute the name of the session library, see “?SLIB? (Session Library) Expression” on page 3-22.

To specify a library, see “SLIB Procedure” on page 4-244 or “LIBRARY OCL Statement” on page 5-54.

To substitute the name of the library used by the System/36 environment to hold data files, see “?FLIB? (Files Library) Expression” on page 3-19.

To substitute the name of the session files library, see “?SFLIB? (Session Files Library) Expression” on page 3-22.

To change the session or current files library, see “FLIB Procedure” on page 4-98 or “FILELIB OCL Statement” on page 5-43.

### Library Existence

To determine whether a library exists on the system, see “DATAF1 (Files and Libraries on Disk) Condition” on page 3-28.

To determine whether a library exists on a diskette, see “DATAI1 (Files on Diskette) Condition” on page 3-29.

To determine whether a library exists on tape or tape cartridge, see “DATAT (Files on Tape) Condition” on page 3-30.

### Member Existence

To determine whether a source member exists in a library, see “SOURCE (Library Source Members) Condition” on page 3-37.

To determine whether a procedure member exists in a library, see “PROC (Library Procedure Members) Condition” on page 3-36.

To determine whether a subroutine member exists in a library, see “SUBR (Library Subroutine Members) Condition” on page 3-38.

To determine whether a load member exists in a library, see “LOAD (Library Load Members) Condition” on page 3-34.

#### **Space Available**

To determine the number of disk blocks available, see “BLOCKS (Available Disk Space) Condition” on page 3-27.

## **Diskette Information**

#### **Volume ID**

To substitute the volume ID of a diskette, see “?VOLID? or ?VOLID'location'? (Diskette or Tape Volume ID) Expression” on page 3-23.

To test for the volume ID of a diskette, see “VOLID (Diskette and Tape Volume IDs) Condition” on page 3-41.

To substitute the device name of the diskette device, see “?DEV'unit'? (Device Name) Expression” on page 3-17.

#### **Existence**

To determine whether a file or library exists on a diskette, see “DATAI1 (Files on Diskette) Condition” on page 3-29.

## **Tape Information**

#### **Volume ID**

To substitute the volume ID of a tape or tape cartridge, see “?VOLID? or ?VOLID'location'? (Diskette or Tape Volume ID) Expression” on page 3-23.

To test for a volume ID of a tape or tape cartridge, see “VOLID (Diskette and Tape Volume IDs) Condition” on page 3-41.

To substitute the device name of a tape device, see “?DEV'unit'? (Device Name) Expression” on page 3-17.

#### **Existence**

To determine whether a file or library exists on a tape or tape cartridge, see “DATAT (Files on Tape) Condition” on page 3-30.

## **Comparing and Evaluating Values and Branching in Procedures**

#### **Comparing**

To test the values of parameters or to test for one or more conditions, see “IF Conditional Expressions” on page 3-25.

To compare one expression with another, see “string1=string2 (Comparing, Equal To) Condition” on page 3-40 or “string1>string2 (Comparing, Greater Than) Condition” on page 3-41.

#### **Evaluating**

To assign a value to a parameter, to add, subtract, multiply, or divide values, to set the ?CD? (return code) expression, or to evaluate substitution expressions, see “EVALUATE Statement” on page 3-46.

#### **Branching**

To branch to another statement in a procedure, see “GOTO and TAG Statements” on page 3-52.

## Testing the Procedure or Job Environment

### Display Station Type

To determine whether a procedure is running from the system console, see “CONSOLE (System Console) Condition” on page 3-27. To determine whether a procedure is running from a display station that can display double-byte characters during an double-byte session, or that can display 27 lines and 132 characters per line, see “DSPLY (Display Station Type) Condition” on page 3-31.

### Job Queue

To determine whether a procedure is being run from the job queue, see “JOBQ (Job Queue) Condition” on page 3-33.

### Evoked Procedure

To determine whether a procedure was started by an EVOKE OCL statement or by the intersystem communications function (OS/400-ICF), hereafter referred to as ICF. See “EVOKE OCL Statement” on page 5-23.

### Inquiry Mode

To determine whether a procedure is run during inquiry mode, see “INQUIRY (Inquiry Mode) Condition” on page 3-33.

### MRT Requestors

To determine whether the maximum number of requester terminals are attached to a multiple requester terminal (MRT) procedure, see “MRTMAX (Multiple Requesting Terminals) Condition” on page 3-35.

### Security Level

To determine whether password security is active on the system, see “SECURITY (Password Security) Condition” on page 3-36.

To determine the security classification of an operator running the procedure, see “SECURITY (Password Security) Condition” on page 3-36.

### Running Procedures

To determine whether one or more specified procedures are currently running on the system, see “ACTIVE (Running Procedures) Condition” on page 3-26.

### Switches

The switches are also called indicators U1 through U8. To determine the switch settings, see “SWITCH (Switches) Condition” on page 3-39.

## Ending Procedures

### End

To cancel a procedure, see “CANCEL Statement” on page 3-46.

To end a nested procedure and return to the calling procedure, see “RETURN Statement” on page 3-55.

To end a procedure and call the same or another procedure, see “RESET Statement” on page 3-54.

---

## \* (Comment) Statement

Comment statements usually explain the purposes of the statements in a procedure. Comments do not display when the procedure runs, only when it prints or displays.

```
* comment
```

59020011-0

**comment** This is any combination of words and characters. Enter the asterisk (\*) in column 1. Any characters following the asterisk (\*) do not process when the procedure runs.

### Example 1

The following example shows a comment statement as the first statement in a procedure.

```
* TESTA procedure
// LOAD PROG1
// FILE NAME-INPUT
// RUN
```

### Example 2

If you have many comment statements, put a RETURN statement at the end of the procedure and put your comments after the RETURN. This way the system processes only the RETURN statement. Your comments do not process (thus saving the amount of time the system otherwise uses to read the comments). For example:

```
// ...
// ... (statements in the procedure)
// ...
// RETURN
*
*
* (comments)
*
*
```

---

## Substitution Expressions

Substitution expressions allow you to substitute information into the statements processed when the procedure runs. Examples of information you can substitute are:

- Positional parameters on the statement that called the procedure.
- Information supplied by the operator in response to a display prompt or a message issued from within the procedure. If a procedure that issues messages is placed on the job queue, the messages appear on the system console, and the system operator must know the correct responses.
- Specified positions in the local data area. (For an explanation of how to place data into the local data area, see “LOCAL OCL Statement” on page 5-56.)

You can use substitution expressions while entering statements from the keyboard or in the command statement in a menu. However, if a substitution expression is used for a *positional* parameter (one of the following expressions: ?n?, ?n'value'?, ?nT'value'?, ?nF'value'?, ?nR?, ?nR'mic'?) from the keyboard or contained in a menu, the result is a null substitution (no value is substituted). A substitution expression for a *nonpositional* parameter results in the proper substitution. For example, you can enter the following statement from the keyboard:

```
// FILE NAME-?2'FILEA'?,DATE-?DATE?
```

If the current date is 021480, the following statement is generated:

```
// FILE NAME-,DATE-021480
```

Comment statements, which are indicated by an asterisk (\*) in position 1 of the statement, are not completely processed by the System/36 environment; therefore, substitution is not performed on comment statements. Substitution occurs whenever a valid expression is encountered, even if the expression is in the comment portion of an OCL statement.

Substitution expressions always begin and end with a question mark. A substitution expression begins any time a question mark is immediately followed by a number or by one of the following letters: C, D, F, L, M, P, R, S, T, U, V, or W. This means that, for example, if you specify ?C in a procedure, it will always process as a substitution expression. An error message can display if you entered the ?C as part of anything else.

However, if a question mark is followed by a valid name of 10 characters or less, it is not treated as a substitution expression. Therefore, you can specify prompting for a CL command (for example, ?CRTCLPGM) without causing an error. A question mark followed by any other letter is not treated as a substitution expression.

If the OS/400 program provides quotes for names, the quotes are not shown when you use a substitution expression. If you use the OS/400 program to create 10-character names, the System/36 environment substitution expressions cannot process them.

Following are descriptions of the substitution expressions that you can use in a procedure. The descriptions include examples of how to use the substitution expressions.

## ?n? (Parameter) Expression

This expression substitutes the value of the nth positional parameter. If the nth parameter does not have a value, no value is substituted.

**n** This is a number from 1 through 64 that specifies the parameter to substitute.

For example, a procedure contains the following statement:

```
// * '?3? WAS DELETED'
```

If the third parameter does not have a value (that is, you did not specify a value on the procedure statement and a previous statement within the procedure did not assign it a value), the following statement is generated:

```
// * ' WAS DELETED'
```

If the value of the third parameter is FILEX, the following statement is generated:

```
// * 'FILEX WAS DELETED'
```

## ?n'value'? (Default Parameter) Expression

This expression substitutes the value of the nth positional parameter. If the nth parameter does not have a value, the expression permanently assigns a default value to the parameter and then substitutes that value.

**n** This is a number from 1 through 64 that specifies the parameter to substitute.

**value** This specifies the value assigned to the parameter if the parameter currently has no value. Any following references to the nth parameter within the procedure use the assigned value.

For example, a procedure contains the following statement:

```
// FILE NAME-?2'FILEA'?
```

If the second parameter does not have a value, the following statement is generated:

```
// FILE NAME-FILEA
```

The following example shows how references to the second parameter that follow the default expression also use the value FILEA:

```
// FILE NAME-?2'FILEA'?  
*  
*  
// FILE NAME-???
```

If the second parameter does not have a value, the following statements are generated:

```
// FILE NAME-FILEA  
*  
*  
// FILE NAME-FILEA
```

## ?nT'value'? (Temporary Value Parameter) Expression

This expression substitutes the value of the nth positional parameter. If the nth parameter does not have a value, the expression temporarily assigns a value to the parameter and then substitutes the value.

**n** This is a number from 1 through 64 that specifies the parameter to substitute.

**T** This specifies that the value is temporarily assigned to the parameter.

**value** This specifies the temporary value. A temporary value is used only for the current substitution expression. For other references to the nth parameter within the procedure, the parameter does not have a value.

For example, a procedure contains the following statements:

```
// FILE NAME-?2T'FILEA'?  
// * 'THE SECOND PARAMETER IS ?2T'NOT DEFINED: FILEA ASSUMED'?'  
// * 'THE SECOND PARAMETER IS NOW ???'
```

If the second parameter does not have a value, the following statements are generated:

```
// FILE NAME-FILEA
// * 'THE SECOND PARAMETER IS NOT DEFINED: FILEA ASSUMED'
// * 'THE SECOND PARAMETER IS NOW '
```

Note that the ?? expression in the last statement substituted the original value of parameter 2, which was blank.

If the second parameter was FILEC, the following statements are generated:

```
// FILE NAME-FILEC
// * 'THE SECOND PARAMETER IS FILEC'
// * 'THE SECOND PARAMETER IS NOW FILEC'
```

## ?nF'value'? (Forced Value Parameter) Expression

This expression immediately forces a new value to be assigned to the nth positional parameter, even if the nth positional parameter already has a value.

- n** This is a number from 1 through 64 that specifies the parameter to substitute.
- F** This indicates that the value is to be assigned to the parameter regardless of the parameter's current value.
- value** This specifies the value that substitutes into the statement containing the expression.

For example, a procedure contains the following statements:

```
// LOAD PROG1
// FILE NAME-INPUT,LABEL-?3F'FILEA'?
// RUN
// LOAD PROG2
// FILE NAME-INPUT,LABEL-?3?
// RUN
```

The following statements are generated, regardless of parameter 3's value before program PROG1 was loaded:

```
// LOAD PROG1
// FILE NAME-INPUT,LABEL-FILEA
// RUN
// LOAD PROG2
// FILE NAME-INPUT,LABEL-FILEA
// RUN
```

To change a parameter so that it no longer has a value, specify the following:

```
?nF' '?
```

## ?R? (Required Parameter) Expression

This expression displays the message:

```
Enter required parameter
```

and waits for the operator to enter (at the keyboard) the value to substitute. R indicates that an operator reply is required. Enter up to 60 characters in response to the message. For example, a procedure contains the following statement:

```
// FILE NAME-?R?
```

When the statement is encountered, the message Enter required parameter displays. The operator then types FILEA at the keyboard, presses Enter, and the following statement is generated:

```
// FILE NAME-FILEA
```

See “Procedure Parameters” on page 5-2 for more information on procedure parameters.

## ?nR? (Missing Parameter) Expression

This expression substitutes the value of the nth positional parameter. If the nth parameter does not have a value, a message displays:

```
Enter missing parameter
```

The expression waits for the operator to enter (at the keyboard) the value to substitute. References to the nth parameter that follow this expression use the value entered by the operator. Enter up to 60 characters in response to the message.

**n** This is a number from 1 through 64 that specifies the parameter to substitute.

**R** This indicates that the parameter is required.

For example, a procedure contains the following statement:

```
// FILE NAME-INPUT,LABEL-?1R?
```

If the first parameter does not have a value, the message Enter missing parameter displays. The operator then enters FILEA from the keyboard, and the following statement is generated:

```
// FILE NAME-INPUT,LABEL-FILEA
```

See “Procedure Parameters” on page 5-2 for more information on procedure parameters.

## ?R‘mic’? (Required Parameter Message) Expression

This expression displays a message from the current user first-level message member and waits for the operator to enter the value to substitute. Enter up to 60 characters in response to the message. See “MEMBER OCL Statement” on page 5-59 for information about assigning a message member to a procedure. See “CREATE Procedure” on page 4-66 for information about creating a message member.

**R** This indicates that the parameter is required.

**mic** This identifies the message identification code of the message that displays.

For example, a procedure contains the following statements:

```
// MEMBER USER1-MESSAGES  
// LOAD PROGA  
// FILE NAME-?R'0015'?  
// RUN
```

When this statement processes, the System/36 environment displays message 0015 from the current first-level message member named MESSAGES:

Enter the name of the file:

The operator then enters the word PAYROLL at the keyboard and the following is generated for the FILE OCL statement:

```
// FILE NAME-PAYROLL
```

See "Procedure Parameters" on page 5-2 for more information on procedure parameters.

## ?nR'mic'? (Missing Parameter Message) Expression

This expression substitutes the value of the nth positional parameter. If the nth parameter does not have a value, the expression displays a message from the current first-level message member and waits for the operator to enter the value to substitute. Enter up to 60 characters in response to the message. References to the nth parameter that follow this expression use the value the operator enters.

See "MEMBER OCL Statement" on page 5-59 for information about assigning a message member to a procedure. See "CREATE Procedure" on page 4-66 for information about creating a message member.

- n** This is a number from 1 through 64 that specifies the parameter to substitute.
- R** This indicates that the parameter is required.
- mic** This identifies the message identification code of the message that displays if the nth parameter does not have a value.

For example, a procedure contains the following statements:

```
// MEMBER USER1-MESSAGES  
// LOAD PROGA  
// FILE NAME-?1R'0015'?  
// RUN  
// * 'FILE ?1? WAS USED'
```

If you do not enter the first parameter, message 0015 from the current first-level message member named MESSAGES displays:

Enter the name of the file:

The operator then enters the word PAYROLL and the following is generated for the FILE OCL statement:

```
// FILE NAME-PAYROLL
```

See "Procedure Parameters" on page 5-2 for more information on procedure parameters.

## ?Cn? (Parameter Length) Expression

This expression substitutes the length of the nth positional parameter. The result of the substitution is a 3-digit number with leading zeros.

- C** This indicates substitution of character length information.
- n** This is a number from 1 through 64 that specifies the parameter substituted.

For example, a procedure contains the following statement:

```
// IF ?C?>8 * '??' HAS MORE THAN 8 CHARACTERS'
```

If parameter 2 has a value of FILEABCDE, the following statement is generated:

```
// IF 009>8 * 'FILEABCDE HAS MORE THAN 8 CHARACTERS'
```

## ?C'value'? (Length) Expression

This expression substitutes the length of the specified value. The result of the substitution is a 3-digit number with leading zeros.

**C** This indicates substitution of character length information.

**value** This specifies the value whose length is substituted. If the value is less than zero, the minus sign is included in the length. For example, -23 has a length of 003.

For example, a procedure contains the following statement:

```
// * '?USER?' HAS ?C'?USER? '?' CHARACTERS'
```

If the operator has a user ID of SUSAN, the following statement is generated:

```
// * 'SUSAN HAS 005 CHARACTERS'
```

## ?CD? (Return Code) Expression

This expression allows you to check conditions in a procedure. It substitutes a 4-digit return code set by the system, by a licensed program, or by a procedure using the EVALUATE statement. Table 3-1 shows the possible return codes.

Table 3-1 (Page 1 of 4). ?CD? Return Code Definitions

Return Code	Meaning
0000	The previous job step ended normally, or this is the first step in the job. This value also returns when you press the Enter key on a PROMPT OCL statement.
1002	The COBOL compilation contains warning errors.
1004	The COBOL compilation contains conditional errors.
1008	The previous job step contains serious errors (BLDMENU, COBOLC, FORMAT, or RPGC procedures).
1010	An operator chose the 0 option in response to the ERR procedure.
1011	An operator chose the 1 option in response to the ERR procedure.
1012	An operator chose the 2 option in response to the ERR procedure.
1312	The previous job step was an MRT program that ended without releasing the display station.
1991	The operator pressed command key 7 and canceled the help procedure.
2001—2024	Command keys 1 through 24 returned from the PROMPT OCL statement display. 2001 means command key 1, 2002 means command key 2, ..., 2024 means command key 24.

---

Table 3-1 (Page 2 of 4). ?CD? Return Code Definitions

---

Return Code	Meaning
2030	<p>On System/36, this return code indicates that the file specified on the FILE OCL statement is not available because it is in use by a suspended program, a program with the never-ending program (NEP) attribute, a job waiting because of a WAIT OCL statement, or a job that has acquired the file using a FILE OCL statement outside a LOAD and RUN OCL statement pair.</p> <p><b>Note:</b> This return code is not supported on the AS/400 system. These conditions are included in return code 2031.</p>
2031	<p>The file specified on the FILE OCL statement is currently in use. Try again to allocate the file. This return code includes conditions for which System/36 returned code 2030.</p>
2032	<p>On System/36, this return code indicates that the diskette drive specified by the ALLOCATE OCL statement is not available. The drive is currently in use by one of the following:</p> <ul style="list-style-type: none"><li>• A suspended program</li><li>• A program with the NEP attribute</li><li>• A job waiting because of a WAIT OCL statement</li><li>• A job that used the ALLOCATE OCL statement to allocate the drive</li></ul> <p><b>Note:</b> This return code is not supported on the AS/400 system. These conditions are included in return code 2033.</p>
2033	<p>The diskette drive specified by the ALLOCATE OCL statement is currently in use. This return code includes conditions for which System/36 returned code 2032.</p>
2034	<p>The CREATE procedure (\$MGBLD utility program) found errors.</p>
2035	<p>On System/36, this return code indicates that tape drive 1 specified by the ALLOCATE OCL statement is not available. This drive is currently in use by one of the following:</p> <ul style="list-style-type: none"><li>• A suspended program</li><li>• A program with the NEP attribute</li><li>• A job waiting because of a WAIT OCL statement</li><li>• A job that used the ALLOCATE OCL statement to allocate the drive</li></ul> <p><b>Note:</b> This return code is not supported on the AS/400 system. These conditions are included in return code 2036.</p>
2036	<p>Tape drive 1 specified by the ALLOCATE OCL statement is currently in use. This return code includes conditions for which System/36 returned code 2035.</p>
2037	<p>On System/36, this return code indicates that tape drive 2 specified by the ALLOCATE OCL statement is not available. The drive is currently in use by one of the following:</p> <ul style="list-style-type: none"><li>• A suspended program</li><li>• A program with the NEP attribute</li><li>• A job waiting because of a WAIT OCL statement</li><li>• A job that used the ALLOCATE OCL statement to allocate the drive</li></ul> <p><b>Note:</b> This return code is not supported on the AS/400 system. These conditions are included in return code 2038.</p>
2038	<p>Tape drive 2 specified by the ALLOCATE OCL statement is currently in use. This return code includes conditions for which System/36 returned code 2037.</p>

---

Table 3-1 (Page 3 of 4). ?CD? Return Code Definitions

Return Code	Meaning
2040	The printer specified in the PRINTER OCL statement is already being continued by the CONTINUE-YES parameter.
2041	On System/36, this return code indicates that the tape cartridge drive specified by the ALLOCATE OCL statement is currently in use by one of the following: <ul style="list-style-type: none"> <li>• A suspended program</li> <li>• A program with the NEP attribute</li> <li>• A job waiting because of a WAIT OCL statement</li> <li>• A job that used the ALLOCATE OCL statement to allocate the drive</li> </ul> <p><b>Note:</b> This return code is not supported on the AS/400 system. These conditions are included in return code 2042.</p>
2042	The tape cartridge drive specified by the ALLOCATE OCL statement is currently in use. This return code includes conditions for which System/36 returned code 2041.
2043	On System/36, this return code indicates that the tape cartridge drive or diskette drive specified by the ALLOCATE OCL statement is not available. A common system resource that waits and cannot be shared is in use by the diskette drive or tape cartridge drive. <p><b>Note:</b> This return code is not supported on the AS/400 system. Return codes 2033 or 2042 are now sent for this condition, depending on the device.</p>
2044	On System/36, this return code indicates that the tape cartridge drive or diskette drive specified by the ALLOCATE OCL statement is not available. A common system resource that cannot be shared is in use by one of the following: <ul style="list-style-type: none"> <li>• The diskette drive</li> <li>• The tape cartridge drive</li> <li>• A suspended program</li> <li>• A program with the NEP attribute</li> <li>• A job waiting because of a WAIT OCL statement</li> <li>• A job that used the ALLOCATE OCL statement to allocate the drive</li> </ul> <p><b>Note:</b> This return code is not supported on the AS/400 system. Return codes 2033 or 2042 are sent for this condition, depending on the device.</p>
2045	The maximum number of requester terminals you can attach to the MRT has been exceeded. Try your request again later.
2090	Page Down key from PROMPT OCL statement.
2091	Page Up key from PROMPT OCL statement.
2092	Help key from PROMPT OCL statement.
2093	Record Backspace key from PROMPT OCL statement. (An operator pressed Home while the cursor was in the home position.)
2143	The operator pressed command key 3 and canceled the help procedure.
3600	Reserved to indicate that a procedure was processed by the SSP operating system that was running on a System/36.
3601	The SYSTYPE procedure was processed by the System/36 environment of the OS/400 operating system.

Table 3-1 (Page 4 of 4). ?CD? Return Code Definitions

Return Code	Meaning
3602	Reserved to indicate that a procedure was processed by the SSP operating system that was running on an AS/400 Advanced 36 SSP only system.
3603	Reserved to indicate that a procedure was processed by the SSP operating system that was running within an AS/400 Advanced 36 machine of the OS/400 operating system.
3721	The operator canceled the previous job step by selecting option 2 in response to a message, or the previous step was an MRT program and the operator released the display station by interrupting the MRT program (by pressing the System Request key) and then selecting option 2.
8158	The OS/400-ICF session ended abnormally. The device received a subsystem error (80 major return code) or session error (81 major return code) from an input/output operation within the previous job step, or the previous step was an MRT job step and the operator canceled the previous step by selecting the 2 option in response to a message.

In the following example, if the return code equals 1008, a message is sent to the display station that started the procedure:

```
// IF ?CD?=1008 MSG ?WS?, 'AN ERROR OCCURRED DURING THE COMPILE'
```

The System/36 environment resets the return code to 0000 whenever it processes a RUN OCL statement.

## ?CLIB? (Current Library) Expression

This expression substitutes the name of the current library. See "LIBRARY OCL Statement" on page 5-54 for more information about the current library. For example, a procedure contains the following statements:

```
// LIBRARY NAME-MYLIB  
// JOBQ ?CLIB?,PROC
```

The following JOBQ statement is generated:

```
// JOBQ MYLIB,PROC
```

The current library is the library that is currently active. The current library is established when you process the LIBRARY OCL statement within a procedure.

## ?DATE? (Program Date) Expression

This expression substitutes the current program date. For example, a procedure contains the following statement:

```
// FILE NAME-FL?DATE?
```

If the current program date is 021483, the following statement is generated. Note that the format is the current session date format:

```
// FILE NAME-FL021483
```

## ?DATEFMT? (Date Format) Expression

This expression substitutes the session date format of MDY, DMY, or YMD.

The following example saves the session date format, changes it, and then restores it to its original value:

```
// EVALUATE P64=?DATEFMT?  
SET ,,DMY  
:  
SET ,,?64?
```

## ?DEV'unit'? (Device Name) Expression

This expression substitutes the 10-character name of the device that is used for the 2-character System/36 environment unit ID specified in 'unit'. If the unit specified is not configured, a null value is substituted.

In the following statement, the device name expression determines if the device T1 is configured:

```
// IF ?DEV'T1'?/ GOTO NOTAPE
```

In the following statement, the name of the work station from which the job runs or was submitted is inserted into an AS/400 command:

```
SBMJOB CMD('CALL ABC') MSGQ(QSYS/?DEV'?WS'??)
```

If the job runs from the work station named WORKST01, the following statement is generated:

```
SBMJOB CMD('CALL ABC') MSGQ(QSYS/WORKST01)
```

## ?F'S,name'? or ?F'S,name,date'? (File Size) Expression

This expression substitutes the number of blocks or records reserved (or allocated) for a database disk file. The value substituted is given in the units (blocks or records) specified when the file was created. The result of the substitution is an 8-digit number with leading zeros. The specified file is located using the library list if the library list search is in effect. If the library list search is not in effect, the file is located in the current files library. Only database files are checked, and file overrides are ignored. 00000000 is substituted in the following situations:

- The file does not exist.
- The file does not have any members.
- The file is allocated exclusively to another job. This happens if the other job refers to the file with a FILE OCL statement that has no DISP parameter, or that has DISP-NEW or DISP-OLD specified.
- The file is a job (RETAIN-J) file.

**Note:** The number of blocks or records reserved is substituted if the files library is QTEMP, or the library list search indicator is set to search the library list and QTEMP is in the library list.

The number 16 777 215 is substituted if the actual number of records is greater than 16 777 215.

If the file has more than one member, but no date is specified, the size of the member with the most recent creation date is substituted.

If the file was created by the CRTPF command rather than by the System/36 environment, the number of records allocated for the file are substituted. If \*NOMAX was specified for the file size when it was created, 08000000 is substituted. If a number greater than 08000000 is specified for the file size at creation time, the number specified is substituted. However, an error can occur if the substitution expression is specified on a FILE OCL statement.

**F** This indicates substitution of file size information.  
**S** This indicates substitution of the allocated size.  
**name** This is the name of the file.  
**date** This is the creation date of the file.

For example, a procedure contains the following statement:

```
// FILE NAME-FILEB,BLOCKS-?F'S,FILEA'?
```

If FILEA was created with a size of 50 blocks, the following statement is generated for FILEB:

```
// FILE NAME-FILEB,BLOCKS-00000050
```

This type of substitution, when used with a FILE OCL statement, could result in an error if the file does not exist.

## **?F'A,name'? or ?F'A,name,date'? (Actual File Size) Expression**

This expression substitutes the actual number of data records in a resident database file. The result of the substitution is an 8-digit number with leading zeros. The specified file is located using the library list if the library list search is in effect. If the library list search is not in effect, the file is located in the current files library. Only database files are checked, and file overrides are ignored. 00000000 is substituted in the following situations:

- The file does not exist.
- The file does not have any members.
- The file is allocated exclusively to another job. This happens if the other job refers to the file with a FILE OCL statement that has no DISP parameter, or that has DISP-NEW or DISP-OLD specified.
- The file is a job (RETAIN-J) file.

**Note:** The actual number of records is substituted if the files library is QTEMP, or the library list search indicator is set to search the library list and QTEMP is in the library list.

The number 16 777 215 is substituted if the actual number of records is greater than 16 777 215.

If the file has more than one member, but no date is specified, the size of the member with the most recent creation date is substituted.

**F** This indicates substitution of file size information.  
**A** This indicates substitution of the actual number of records.  
**name** This is the name of the file.  
**date** This is the creation date of the file.

For example, a procedure contains the following statement:

```
// FILE NAME-FILEB,RECORDS-?F'A,FILEA'?
```

If FILEA actually contains 150 data records, the following statement is generated for FILEB, which creates FILEB with a size of 150 records:

```
// FILE NAME-FILEB,RECORDS-00000150
```

These types of substitutions, when used with a FILE OCL statement, could result in an error if the file does not exist or if the file is empty.

If the file is a remote file, the number of records substituted does not reflect any records added by a user currently using the file.

## ?F'T,name'? (File Type) Expression

This expression substitutes the file organization or the file type of the file specified. The file organization is passed back as either I,D,S, or X, if the file was created by the System/36 environment. If the file was not created by the System/36 environment, the file type is passed back as either PHY, LGL, or SAV. A null value is substituted in the following conditions:

- The file does not exist.
- The file is a job (RETAIN-J) file.

**Note:** The file organization or file type is substituted if the files library is QTEMP, or the library list search indicator is set to search the library list and QTEMP is in the library list.

- The file is not a database data file nor a save file.

**F** This indicates substitution of file information.

**T** This indicates substitution of the file type.

**name** This is the name of the file.

## ?FLIB? (Files Library) Expression

This expression substitutes the 10-character name of the current files library in use by the System/36 environment. For example, a procedure contains the following statement:

```
OVRDBF FILE(TEMP1) TOFILE(?FLIB?/TEMP1)
```

If the name of the files library currently used by the System/36 environment is S36FILES, the following statement is generated:

```
OVRDBF FILE(TEMP1) TOFILE(S36FILES/TEMP1)
```

## ?L'position,length'? (Local Data Area) Expression

This expression substitutes a value from the 512-byte display station local data area. For information on changing the display station local data area, see "LOCAL OCL Statement" on page 5-56. The data substitutes from either the user or system local data area, depending on the last AREA parameter of the LOCAL OCL statement. If you have not previously specified an AREA parameter, the user local data area is assumed. The system local data area is used by IBM-supplied procedures, and you can lose any data you place in the system local data area. If you call an IBM-supplied procedure that uses the system local data area, the data in

your user area is not affected. Also, the local data area in use when you call an IBM-supplied procedure is recognized as the local data area to use when the called procedure ends.

- L** This indicates substitution of a value from the local data area.
- position** This specifies the starting position (a number from 1 through 512, where 1 specifies the first position) of the local data area to substitute.
- length** This specifies the number of positions in the local data area to substitute. Beginning blanks and embedded blanks (blanks inside the characters being substituted, for example, 'ABC DEF') are allowed in the substituted value, but trailing blanks are not substituted.

For example, a procedure contains the following statement:

```
// FILE NAME-INPUT,LABEL-?L'12,8'?,UNIT-F1
```

If eight positions, from 12 through 19, of the local data area contain 'FILEA ', the following statement is generated:

```
// FILE NAME-INPUT,LABEL-FILEA,UNIT-F1
```

If the substituted data is then used in an expression or statement that has a length restriction, the data must obey that restriction.

**Warning:** If the local data area contains double-byte character set (DBCS) data, make sure that an equal number of shift-in and shift-out characters are substituted. Otherwise, the results are unpredictable.

## ?Mmic? or ?M'mic,position,length'? (Message Member) Expression

This expression substitutes a value from a statement in the first-level (USER1) message member. For information about assigning a message member, see "MEMBER OCL Statement" on page 5-59.

- M** This indicates substitution of a value from a statement in the message member.
- mic** This specifies the message identification code of the message containing the value to substitute.
- position** This specifies the starting position (a decimal number from 1 through 75, where 1 specifies the first position) of the message text to substitute.
- length** This specifies the decimal number of positions of the value to substitute.

If you do not specify position and length, the entire message text substitutes. Beginning blanks and embedded blanks (blanks inside the characters being substituted, for example, 'ABC DEF') are allowed in the substituted value, but trailing blanks are truncated. For example, a procedure contains the following statement:

```
// * '?M'0014,10,8'? WAS USED'
```

If eight positions, from 10 through 17, of message 0014 contain 'FILEA ', the following statement is generated:

```
// * 'FILEA WAS USED'
```

If the substituted data is then used in an expression or statement that has a length restriction, the data must obey that restriction.

**Warning:** If the message contains double-byte character set (DBCS) data make sure that an equal number of shift-in and shift-out characters substitute. Otherwise, the results are unpredictable.

For information about retrieving DBCS messages from the message member, see “Considerations for the Double-Byte Character Set Version of the OS/400 Program” on page 4-70.

## ?MENU? (Current Menu) Expression

This expression substitutes the 1- through 6-character menu name of the currently active user menu. If no menu is currently active, no value substitutes.

For example, a procedure contains the following statement:

```
// IF ?MENU?=SAMPLE MENU MINE
```

If the menu named SAMPLE is currently active, the test is true and the MENU OCL statement processes to display the menu MINE. If the menu was not named SAMPLE, the MENU statement does not process.

Menu names that have quotes provided by the system do not have the quotes shown by the ?MENU? expression.

## ?MSGID? (Message ID) Expression

This expression substitutes the message ID of the error message sent by the last OS/400 command or program run with the LOAD/RUN OCL statements that processed in the job. If no OS/400 command or program was processed previously, or if the last command or program ended with no error message, a null value is substituted. A null value will also be substituted if the CHGS36MSGL command was used to specify a default action of \*IGNORE for the message sent by an AS/400 system command. See “Handling Errors on CL Commands” on page 2-22 for more information. The following example shows how to use the message ID expression with the Check Object (CHKOBJ) command to determine if a particular object exists:

```
CHKOBJ OBJ(?FLIB?/TEMP1) OBJTYPE(*DTAARA)  
// IF ?MSGID?/CPF9801 CRTDTAARA DTAARA(?FILE?/TEMP1) TYPE(*CHAR) LEN(50)
```

## ?PRINTER? (Session Printer) Expression

This expression substitutes the 2-character value that indicates the session printer. The SET or PRINT procedure is used to set the session printer. See “SET Procedure” on page 4-235 or “PRINT Procedure” on page 4-178.

For example, a procedure contains the following statement:

```
// SYSLIST ?PRINTER?
```

If the printer ID of the session printer is P2, the following statement is generated:

```
// SYSLIST P2
```

If the session printer is defined as SYSTEM, the actual printer ID of the system printer substitutes. If the session printer is SYSTEM and no system printer is defined, the word SYSTEM substitutes.

## ?PROC? (First Level Procedure) Expression

This expression substitutes the 1- to 8-character name of the first-level procedure running. The first-level procedure is the first procedure called in a series of nested procedures. For example, a procedure contains the following statement:

```
// * '?PROC? IS RUNNING'
```

If the name of the first level procedure is PROCABC, the following statement is generated:

```
// * 'PROCABC IS RUNNING'
```

Procedure names that have quotes provided by the system do not have the quotes shown by the ?PROC? expression.

If you create a procedure with a name of more than 8 characters, the System/36 environment cannot process it.

## ?SFLIB? (Session Files Library) Expression

This expression retrieves the name of the session files library that will be in use when the current System/36 environment job (that is, the outermost procedure) ends.

## ?SLIB? (Session Library) Expression

This expression substitutes the name of the active library for the session. For example, the session library is MYLIB, and a procedure contains the following statement:

```
// JOBQ ?SLIB?,PROCEDUR
```

The following statement is generated:

```
// JOBQ MYLIB,PROCEDUR
```

The session library is the library that is active at the keyboard. The session library is established when you do one of the following:

- Sign on to the system.
- Run the SLIB procedure.
- Enter the LIBRARY OCL statement at the keyboard.
- Enter the MENU command with a library name.

Library names that have quotes provided by the system do not have the quotes shown by the ?SLIB? expression.

If you create a library with a name of more than 8 characters, the System/36 environment cannot process it.

## ?SYSLIST? (System List Device) Expression

This expression substitutes the 2- to 3-character value that indicates where the system list output displays or prints. Use the SYSLIST or PRINT procedure to set the system list device, see "SYSLIST Procedure" on page 4-253 or "PRINT Procedure" on page 4-178. The following values substitute:

**OFF** Substitutes if the current system list device is set to off.

**CRT** Substitutes if the system list output is set to CRT.

**printer id** (A 2-character printer ID) Substitutes if the system list output is set to a printer.

For example, a procedure contains the following statement:

```
// IF ?SYSLIST?=CRT SYSLIST ?PRINTER?
```

If the system list device is CRT, and if the session printer is P2, the following statement is generated:

```
// IF CRT=CRT SYSLIST P2
```

If you define the system list device as SYSTEM and there is no system printer defined, CRT substitutes.

## **?TIME? (System Time) Expression**

This expression substitutes the current system time in the format HHMMSS, where HH is the hours, MM is the minutes, and SS is the seconds. The system time is taken from the QTIME system value. For example, a procedure contains the following statement:

```
// * 'THE TIME IS ?TIME?'
```

If the current system time is 9:45:24, the following statement is generated:

```
// * 'THE TIME IS 094524'
```

## **?USER? (Operator's User ID) Expression**

This expression substitutes the 1- to 10-character user ID assigned to the operator who started the job. For example, a procedure contains the following statement:

```
// LOCAL DATA- '?USER?'
```

If the user ID of the operator is MIKE, the following statement is generated:

```
// LOCAL DATA- 'MIKE'
```

For MRT procedures, blanks substitute for the ?USER? expression.

**Note:** Avoid using ?USER? to begin a file name if the first character of the user ID to substitute is numeric because the first character of a file name cannot be numeric.

## **?VOLID? or ?VOLID'location'? (Diskette or Tape Volume ID) Expression**

This expression substitutes the 1- to 6-character volume ID of the diskette or tape in the specified location. Use this expression with caution. Otherwise you can bypass the normal volume ID checking meant to ensure that the proper diskette or tape is in use.

**location** This specifies the diskette location or tape unit to search for the diskettes or tapes. Diskette locations are allowed for System/36 compatibility only. If a diskette location is specified, only syntax checking is performed. The device defined in your System/36 environment configuration is searched. Specify any of the following:

<b>S1</b>	This specifies the volume ID of the diskette in slot 1. If you did not enter a location, this location is assumed.
<b>S2</b>	This specifies the volume ID of the diskette in slot 2.
<b>S3</b>	This specifies the volume ID of the diskette in slot 3.
<b>M1.nn</b>	This specifies the volume ID of the diskette in magazine 1, position nn. <b>nn</b> is any number from 01 through 10.
<b>M2.nn</b>	This specifies the volume ID of the diskette in magazine 2, position nn. <b>nn</b> is any number from 01 through 10.
<b>T1</b>	This specifies the volume ID of the tape on tape drive 1.
<b>T2</b>	This specifies the volume ID of the tape on tape drive 2.
<b>TC</b>	This specifies the volume ID of the tape in the tape cartridge drive.

For example, a procedure contains the following statement:

```
// * 'THE VOLUME ID OF THE DISKETTE IS ?VOLID'M1.02'?
```

If the volume ID of the diskette in the diskette drive is VOL001, the following statement is generated:

```
// * 'THE VOLUME ID OF THE DISKETTE IS VOL001'
```

## ?WS? (Display Station ID) Expression

This expression substitutes the 2-character work station ID of the display station that called the procedure. For example, a procedure contains the following statement:

```
// FILE NAME-INPUT,LABEL-FILEA?WS?
```

If display station W3 calls the procedure, the following statement is generated:

```
// FILE NAME-INPUT,LABEL-FILEAW3
```

If the procedure runs from the job queue using the JOBQ command or OCL statement, the System/36 environment substitutes the ID of the display station from which the job was placed on the queue.

If the procedure runs using the EVOKE OCL statement, the System/36 environment substitutes the ID of the display station from the job that started the evoke process.

If the procedure runs in a batch job started by a CL command, such as SBMJOB, the System/36 environment substitutes #0.

---

## Nested Substitution Expressions

You can nest substitution expressions (that is, have substitution expressions that call other substitution expressions). For example, a procedure contains the following statement:

```
?M'0014,1,29'?
```

This expression specifies substitution of the first 29 characters of message number 0014. In this example, the first 29 characters of message 0014 are:

```
// FILE NAME-FILEA?WS?
```

Therefore, the statement that results from the first substitution expression contains another substitution expression. The System/36 environment then processes the latter expression. If the work station ID of the display station that started the job is W1, the resulting statement is:

```
// FILE NAME-FILEAW1
```

By using nested expressions, you can use a substitution expression as a default value within another expression. For example:

```
?1'?2?'
```

instructs the System/36 environment to substitute the value of the first parameter. If the first parameter does not have a value, the System/36 environment substitutes the value of the second parameter.

When using nested substitution expressions, take care to avoid expressions that might cause an infinite loop. For example, a procedure contains the following statement:

```
// FILE NAME-?1R?
```

If the operator did not specify the first parameter, a message displays telling the operator to enter the first parameter value. If the operator enters ?1?, the System/36 environment *continually* substitutes ?1? into the statement. The operator must interrupt and then cancel the procedure to stop the substitution.

You can have an unlimited number of nested substitution expressions in any one statement.

---

## IF Conditional Expressions

Conditional expressions within a procedure allow you to conditionally process certain OCL and utility control statements when the procedure runs.

You can only use the IF expression within a procedure. An IF expression tests for a specified condition. If the condition is met, the specified statement processes.

IF expressions can have any of the following formats:

```
// IF  condition  statement
Or:
// IFT condition  statement
Or:
// IFF condition  statement
```

59020012-0

IF and IFT test whether the condition specified by **condition** is true. If the specified condition is true, the statement following the condition processes.

IFF tests whether the condition specified by **condition** is false (if not true). If the specified condition is false, the statement following the condition processes.

## ACTIVE Condition

The **condition** portion of the IF expression must be a continuous string of nonblank characters. A blank indicates the end of the condition. Blanks between apostrophes are allowed, for example 'AB D'. The conditions you can specify are shown beginning with "ACTIVE (Running Procedures) Condition" on page 3-26.

An IF or IFT test for multiple conditional expressions occurs when a multiple condition is met. For example:

```
// IF ACTIVE-'PROC1,PROC2' PROC3
```

Procedure PROC3 runs if PROC1, PROC2, or both PROC1 and PROC2 are running on the system.

An IFF test for multiple conditional expressions is false when all of the multiple conditions are met. For example:

```
// IFF ACTIVE-'PROC1,PROC2' PROC3
```

Procedure PROC3 runs if both PROC1 and PROC2 are not running on the system.

The **statement** portion of the IF expression is described in "Statement Portion of IF Conditional Expression" on page 3-43.

You can also use an ELSE expression to do a function when the IF expression is not satisfied. See "ELSE Expressions" on page 3-43 for additional information.

When you are using certain conditional expressions, the specified condition may be true *at the time* the condition is tested. However, it may not be true when you try to process something later. Do processing based upon the results of an IF expression with care because the condition being tested for may have changed.

## ACTIVE (Running Procedures) Condition

The ACTIVE conditional expression determines whether one or more procedures are running on the system.

If the condition is true, at least one of the specified procedures was active *at the time* the ACTIVE test was evaluated. Do processing based upon the ACTIVE test with caution because the condition being tested for may have changed. If you try to test for the procedure that contained the ACTIVE test, the test is always false. You cannot use the ACTIVE test to determine if you are the last user of an MRT.

If you test an NRT, the procedure name must be the procedure that contains the // RUN statement of the NRT.

```
// { IF } ACTIVE- { procedure name } statement  
   { IFT } { 'proc1,proc2,...,procn' }  
   { IFF }
```

S9020013-0

### procedure name

This specifies the name of a procedure. The condition is true if the specified procedure is currently running on the system.

### 'proc1,proc2,...,procn'

This specifies two or more procedure names. For IF and IFT tests, the condition is true if any of the specified procedures are currently running

on the system. For IFF tests, the condition is false if all the specified procedures are not running on the system.

**statement**

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**Example 1**

This example specifies that if a procedure called PROC1 is active, the system should cancel the running of the procedure that contains the IF test.

```
// IF ACTIVE-PROC1 CANCEL
```

**Example 2**

This example specifies the running of the procedure PAYROLL only if both procedure PROC1 and procedure PROC2 are not active.

```
// IFF ACTIVE-'PROC1,PROC2' PAYROLL
```

## BLOCKS (Available Disk Space) Condition

The BLOCKS conditional expression determines whether the specified amount of disk space is available.

If the condition is true, the specified number of blocks was available *at the time* the BLOCKS expression was evaluated. If other programs are running, another program might use that available space before your program tries to use it.

```
// { IF } BLOCKS-size statement
   { IFT }
   { IFF }
```

59020014-0

**size** This specifies the number of blocks, and can be a 1- to 8-digit number. This condition is always true if the specified number of blocks is available before the auxiliary storage threshold is exceeded.

**statement**

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**Example**

This example specifies that if 150 disk blocks are not available, the job should cancel.

```
// IFF BLOCKS-150 CANCEL
```

## CONSOLE (System Console) Condition

The CONSOLE conditional expression checks the QCONSOLE system value to determine whether the procedure is running from the system console.

```
// { IF } CONSOLE- { YES } statement
   { IFT }          { NO }
   { IFF }
```

59020015-0

## DATAF1 Condition

- YES** This is true if the procedure is running from the system console.
- NO** This is true if the procedure is not running from the system console.
- statement** This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

### Example

This example specifies that if the procedure is not running on the system console, the system should cancel the running of the procedure.

```
// IF CONSOLE-NO CANCEL
```

## DATAF1 (Files and Libraries on Disk) Condition

The DATAF1 conditional expression determines whether the specified file exists on disk in the current files library or whether the specified library exists on disk. This condition is true if a resident (T) file or library exists on a disk with the name and creation date (optional) specified. The condition is false if a database file with the specified name exists but does not have any members. This conditional expression does *not* detect a scratch (S) or job (J) file with the name specified. However, if the files library is set to QTEMP, or the library list search indicator is set to search the library list and QTEMP is in the library list, then a scratch (S) or (J) file with the name specified is detected.

If you specify ALL, the condition is true if a file or library exists on the disk that starts with the characters in the specified name.

```
// { IF } DATAF1- { name } statement
   { IPT }      { 'name,date' }
   { IFF }
```

SP020016-0

- name** This specifies the name of the file or library to search for.
- date** This specifies the creation date of the file. The date is optional. If you do not specify the date, the file having the most recent creation date is assumed. If you specify the date, it must be in the session date format. If you specify the name of a library, do not specify the date. If you specify the name of a library and the date, a false condition returns. Enter the date in one of the following formats: mmddyy, ddmmyy, or yymmdd.
- ALL** This specifies searching of the disk for a file or library that starts with the characters in the specified name. If no files or libraries that start with the given characters exist, a false condition returns. If you specify the ALL parameter, only local files and libraries are detected. If you do not specify the ALL parameter, the network resource directory is searched for the file or library.
- statement** This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

### Example

This example specifies that if a disk file with the name specified by the second positional parameter does not exist, the BLDFILE procedure should run to create the file.

```
// IFF DATAF1-??? BLDFILE ???,S,BLOCKS,10,100
```

## DATAI1 (Files on Diskette) Condition

The DATAI1 conditional expression determines whether the specified file exists on diskette. The condition is true if a file exists on the diskette with the name and creation date (optional) specified.

```
// { IF } DATAI1- { name } statement
   { IFT }      { 'name,date,location' }
   { IFF }      { 'name,,location' }
```

59020475-0

- name** This specifies the name of the file searched for.
- date** This specifies the creation date of the file. The date is optional. If you do not specify the date, the test is true when the first file is found. Enter the date in any of the following formats: mmddy, ddmyy, or yymmdd.
- location** This specifies the diskette location. The device defined in your System/36 environment configuration is searched. This parameter is allowed for System/36 compatibility, but only syntax checking is performed. Enter any of the following values:
- S1, S2, or S3** This identifies an individual diskette slot. If you do not specify a location, S1 is assumed.
  - ALLS** This specifies searching of all slots (S1, S2, and S3).
  - M1.nn** This identifies a location within magazine 1. **nn** is any number from 01 through 10. For example, M1.04 indicates location 4 within magazine 1.
  - M2.nn** This identifies a location within magazine 2. **nn** is any number from 01 through 10.
  - ALL1** This specifies searching of all diskettes in magazine 1.
  - ALL2** This specifies searching of all diskettes in magazine 2.
  - ALL** This specifies searching all diskettes in both magazines.
- statement** This specifies a statement to process. See "Statement Portion of IF Conditional Expression" on page 3-43 for more information.

### Example 1

This example specifies that if a file named FILEA exists on the diskette, the DELETE procedure runs, deleting that file.

```
// IF DATAI1-FILEA DELETE FILEA,I1
```

### Example 2

This example specifies that if a file named FILEB does not exist on the diskette, the SAVE procedure runs, copying the file to the diskette.

```
// IFF DATA11-FILEB SAVE FILEB,, ,VOL001
```

## DATAT (Files on Tape) Condition

The DATAT conditional expression determines whether the specified file exists on tape. The condition is true if a file exists on the specified tape drive with the name and creation date (optional) specified.

If a tape is not mounted on the specified unit or if the specified unit is offline, an error message is issued.

```
// { IF } DATAT- { name } statement  
   { IFT } { 'name, date, unit, end' }  
   { IFF }
```

59020017-0

- name** This specifies the name of the file searched for.
- date** This specifies the creation date of the file. The date is optional. If you do not specify the date, the test is true when the first file with the specified name is found. Enter the date in any one of the following formats: mmddy, ddmmy, or yymmdd.
- unit** This specifies the tape unit checked if the system has more than one tape drive. It is any of the following:
- T1** This checks the tape mounted on tape drive 1. If you do not specify unit, T1 is assumed.
  - T2** This checks the tape mounted on drive 2.
  - TC** This checks the tape in the tape cartridge drive.
- end** This specifies the position of the tape after the tape is searched. The position of the tape can be any of the following:
- REWIND** If reel-to-reel is specified, this specifies that the tape is rewound to the load point after processing. If tape cartridge is specified, the tape is positioned to the beginning of the cartridge after processing completes.
  - LEAVE** If reel-to-reel or tape cartridge is specified, this specifies that the tape is left where it was last processed. The next operation to the tape begins at that point.
  - UNLOAD** If reel-to-reel tape is specified, this specifies that the tape is rewound and unloaded after processing completes. If tape cartridge is specified, the tape is positioned to the end of the cartridge after processing completes.
- Each time a new or different tape cartridge is processed, or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation of the cartridge. To prepare a cartridge for operation, the cartridge is wound to the end of the tape, then rewound to the beginning. If REWIND or LEAVE is specified for the last operation, this

can be a lengthy process. The preparation time for the tape can be shortened if you specify UNLOAD, because the tape is left at the end of the cartridge after the last operation completes.

**statement** This specifies the statement processed. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**Example 1**

This example specifies that if a file named FILEA does not exist on tape drive 1, the SAVE procedure runs, copying the file to the end of the tape.

```
// IFF DATAT-'FILEA,,T1,LEAVE' SAVE FILEA,,VOL001,T1,,,,,LEAVE
```

**Example 2**

This example specifies that if a file named FILEB does exist on tape drive 2, the RESTORE procedure runs, copying the file from tape to disk.

```
// IF DATAT-'FILEB,,T2,REWIND' RESTORE FILEB,,,,,T2
```

**DSPLY (Display Station Type) Condition**

The DSPLY conditional expression determines the type of display station in use.

```
// { IF } DSPLY- { 1920 } statement
   { IFT }      { IGC }
   { IFF }      { 24X80 }
                                     { 27X132 }
```

59020018-0

- 1920** This is supported for compatibility with the IBM System/34. This test is always true.
- IGC** This is valid for the double-byte version of the OS/400 program. This test is true if the procedure containing the expression is running during a double-byte character set (DBCS) session at a display station that can display DBCS characters.
- 24X80** This is the same as specifying DSPLY-1920. This test is always true. It means that the procedure is running at a display station that can display 24 lines and 80 characters per line.
- 27X132** This is true if the procedure is running at a display station that can display 27 lines and 132 characters per line. The 3180 Model 2 display station can switch back and forth between this 27-line by 132-character display and a 24-line by 80-character display.

**statement** This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**Example**

This example specifies that during a double-byte character set (DBCS) session, if the display station from which this expression runs is a DBCS-capable display station, then the PROMPT OCL statement displays a format called FMT1 in a load member called FORMATS.

```
// IF DSPLY-IGC PROMPT MEMBER-FORMATS,FORMAT-FMT1
```

## ENABLED (Enabled Communications) Condition

Not supported. On System/36, the ENABLED conditional expression determined whether the specified configuration was enabled.

To do a similar function, use the Retrieve Configuration Status (RTVCFGSTS) command. The RTVCFGSTS command allows you to retrieve configuration status from three configuration objects:

- Line
- Controller
- Device

See the *CL Reference* book for more information on the RTVCFGSTS command.

## EVOKED (Evoked Procedures) Condition

The EVOKED conditional expression determines whether an ICF evoke or the EVOKE OCL statement started a procedure.

```
// { IF } EVOKED- { YES } statement  
   { IFT }  
   { IFF }
```

59020020-0

**YES** This is true if the procedure is evoked through ICF or by the EVOKE OCL statement.

**NO** This is true if the procedure is evoked through ICF or by the EVOKE OCL statement.

### **statement**

This specifies a statement to process. See the “Statement Portion of IF Conditional Expression” on page 3-43.

### **Example 1**

This example specifies that if the procedure containing this statement was evoked by the EVOKE OCL statement, the message Procedure running is sent to the display station that did the evoke operation.

```
// IF EVOKED-YES MSG ?WS?,Procedure running
```

### **Example 2**

This example specifies that if the procedure containing this statement was not evoked by the EVOKE OCL statement and is not running from the job queue, the message Procedure running displays at the display station.

```
// IF JOBQ-NO IF EVOKED-NO * 'Procedure running'
```

## FILELIBL (File Library List) Condition

The FILELIBL conditional expression determines whether existing files should be located through the library list of the job, based on the current library list search indicator.

```
// { IF } FILELIBL- { YES } statement
   { IFT }
   { IFF }
```

RV2P185-1

**YES** This is true if the current library list search indicator is set to indicate that the library list should be used to search for existing files.

**NO** This is true if the current library list search indicator is set to indicate that only the current files library should be used to search for existing files.

**statement**

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**INQUIRY (Inquiry Mode) Condition**

The INQUIRY conditional expression determines whether a procedure is running under inquiry mode. That is, the operator used the System Request 1 option or the TFRSECJOB command to start a second job at this work station. When both jobs are active at the work station, they are considered to be running in inquiry mode.

```
// { IF } INQUIRY- { YES } statement
   { IFT }
   { IFF }
```

59020021-0

**YES** This is true if the procedure containing the expression is running in inquiry mode.

**NO** This is true if the procedure is not running under inquiry mode.

**statement** This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**Example**

This example specifies that if the procedure containing this statement is running in inquiry mode, the procedure is canceled.

```
// IF INQUIRY-YES CANCEL
```

**JOBQ (Job Queue) Condition**

The JOBQ conditional expression determines whether a procedure is run from the job queue.

```
// { IF } JOBQ- { YES } statement
   { IFT }
   { IFF }
```

59020022-0

**YES** This is true if the procedure is running from the job queue.

**NO** This is true if the procedure is not running from the job queue.

## LOAD Condition

**statement** This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

### Example 1

This example specifies that if the procedure containing the statement is running from the job queue, the message Procedure running is sent to the display station that placed the job on the job queue.

```
// IF JOBQ=YES MSG ?WS?, 'Procedure running'
```

### Example 2

This example specifies that if the procedure containing this statement is not running from the job queue, the message Procedure running displays.

```
// IF JOBQ=NO * 'Procedure running'
```

## LISTDONE (Phone List Completion) Condition

Not supported. On System/36, the LISTDONE conditional expression determined whether all the numbers in a phone list had been called.

No similar function is available on the AS/400 system. The OS/400 program does not support phone lists. You can change the connection number associated with a controller description by using the appropriate change controller description command. For example, to change the connection number for an APPC controller, use the Change Controller Description (APPC) (CHGCTLAPPC) command, specifying the new phone number for the connection number (C>NNNBR) parameter. This allows you to call a list of phone numbers in either of two ways:

- **Manually.** Repeat the following steps for each phone number to be called:
  1. Use the appropriate change controller description command, specifying the phone number to be called for the connection number (C>NNNBR) parameter.
  2. Run your communications program.
- **From a program.** Write a program to process a list of phone numbers from a database file. Do the following for each number on the list:
  1. Call a command language (CL) program that runs the appropriate change controller description command, specifying the phone number to be called for the C>NNNBR parameter.
  2. Call the associated communications program.

See the *CL Reference* book for more information on the change controller description (CHGCTLxxx) commands. See the *System/36 Environment Programming* book for more information on using communications in the System/36 environment. See the *CL Programming* book for more information on writing CL programs.

## LOAD (Library Load Members) Condition

The LOAD conditional expression determines whether a library load member exists in a specified library. The condition is true if the specified load member exists in the specified library.

A LOAD member can be any of the following OS/400 object types:

- \*PGM (program)

- \*MSGF (message file)
- \*FILE (display file)

**Note:** A display file has a subtype of \*DSPF.

```
// { IF } LOAD- { member name } statement
   { IFT }   { 'member name,library name' }
   { IFF }
```

59020024-0

#### member name

This specifies the name of the library load member to search for.

#### library name

This specifies the name of the library in which to search for the load member. If you do not specify a library name, the system library (#LIBRARY) is assumed.

#### statement

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

#### Example

This example specifies that if a load member named PROG1 exists in the current library (indicated by ?CLIB?), the REMOVE procedure is run to remove that member.

```
// IF LOAD-'PROG1,?CLIB?' REMOVE PROG1,LOAD,?CLIB?
```

## MRTMAX (Multiple Requesting Terminals) Condition

The MRTMAX conditional expression determines whether a multiple requester terminal (MRT) procedure has the maximum number of requester terminals. The condition is true if the specified MRT procedure has the maximum number of requesters attached.

```
// { IF } MRTMAX-procedure name statement
   { IFT }
   { IFF }
```

59020025-0

#### procedure name

This specifies the name of a MRT procedure.

#### statement

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

#### Example

This example specifies that if the ORDERS procedure has the maximum number of users attached, two messages display, and a PAUSE statement and a CANCEL statement process.

## SECURITY Condition

```
// IFF MRTMAX-ORDERS GOTO START
// * 'Too many people are using the ordering procedure'
// * 'Canceling procedure; Try again later'
// PAUSE
// CANCEL
// TAG START
ORDERS
```

## PROC (Library Procedure Members) Condition

The PROC conditional expression determines whether a library procedure member exists in a specified library. The condition is true if the specified procedure member exists in the specified library.

```
// { IF } PROC- { member name } statement
   { IFT }   { 'member name,library name' }
   { IFF }
```

59020026-0

### member name

This specifies the name of the library procedure member to search for.

### library name

This specifies the name of the library in which to search for the procedure member. If you do not specify a library name, the system library (#LIBRARY) is assumed.

### statement

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**Note:** All procedures are members of the QS36PRC source file in the specified library.

### Example

This example specifies that if a procedure member named TEST exists in the current library (indicated by ?CLIB?), the REMOVE procedure runs, to remove that member.

```
// IF PROC-'TEST,?CLIB?' REMOVE TEST,PROC,?CLIB?
```

## SECURITY (Password Security) Condition

The SECURITY conditional expression determines the following:

- Whether password security is active on the system
- The operator's security classification

```
// { IF } SECURITY- { ACTIVE } statement
   { IFT }         { M }
   { IFF }         { S }
                   { O }
                   { C }
                   { D }
```

59020027-0

**ACTIVE** This is true if password security is active on the system. Password security is active if the QSECURITY system value is 20 or 30.

**M, S, O, C, and D**

These specify security levels. The condition is true if the operator's security classification equals or exceeds the specified security classification. If password security is not active, all tests are true.

The following list shows the available security levels. The value in parentheses is the equivalent user class value in the OS/400 program.

**M** Master security officer (\*SECOFR)

**S** Security officer (\*SECADM)

**O** System operator (\*SYSOPR)

**C** Subconsole operator (\*SYSOPR)

**D** Display station operator (\*USER)

**statement**

This specifies a statement to process. See "Statement Portion of IF Conditional Expression" on page 3-43 for more information.

**Example 1**

This example specifies that if password security is not active, the message is sent to the system console.

```
// IFF SECURITY-ACTIVE ** 'PROCEDURE ?PROC? BEING RUN BY ?USER?'
```

**Example 2**

This example specifies that this procedure can only run if the operator has a security level of system operator, security officer, or master security officer. If the operator has a security level of subconsole operator or display station operator, the procedure is canceled. If password security is not active, the procedure must run from the system console.

```
// IFF SECURITY=0 CANCEL
//     ELSE IFF SECURITY-ACTIVE IF CONSOLE-NO CANCEL
// LOAD PROGRAM1
// RUN
```

**SOURCE (Library Source Members) Condition**

The SOURCE conditional expression determines whether a library source member exists in a specified library. The condition is true if the specified source member exists in the specified library.

```
// { IF } SOURCE-{ member name } statement
// { IFT } { 'member name, library name' }
// { IFF }
```

59020028-0

**member name**

This specifies the name of the library source member to search for.

**library name**

This specifies the name of the library in which to search for the source member. If you do not specify a library name, the system library (#LIBRARY) is assumed.

## SUBR Condition

### statement

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**Note:** All source members are members of the QS36SRC source file in the specified library.

### Example

This example specifies that if a source member named FORMATS does not exist in the current library (indicated by ?CLIB?), the SDA procedure runs to create that member.

```
// IFF SOURCE-'FORMATS,?CLIB?' SDA FORMATS,?CLIB?
```

## SUBR (Library Subroutine Members) Condition

The SUBR conditional expression determines whether a program exists in a specified library. The condition is true if the specified program exists in the specified library.

```
// { IF } SUBR- { member name } statement  
   { IFT }      { 'member name,library name' }  
   { IFF }
```

S9020029-0

### member name

This specifies the name of the program member to search for.

### library name

This specifies the name of the library in which to search for the program. If you do not specify a library name, the system (#LIBRARY) is assumed.

### statement

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

### Notes:

1. In the System/36 environment, a subroutine is equivalent to a program (\*PGM) object.
2. The SUBR conditional expression cannot be used to determine if the library contains a query definition.

### Example

This example specifies that if a program member named PROG1 exists in the current library (indicated by ?CLIB?), the REMOVE procedure runs to remove that member.

```
// IF SUBR-'PROG1,?CLIB?' REMOVE PROG1,SUBR,?CLIB?
```

## SWITCH (Switches) Condition

The SWITCH conditional expression determines the settings of the user program status indicator (UPSI) switches. The condition is true if all eight of the switches for the display station are in the specified state.

```
// { IF } SWITCH-switch settings statement
   { IFT }
   { IFF }
```

59020030-0

### switch settings

These consist of 8 characters, one for each of the eight switches (1 through 8). For each of the eight positions in the switch settings value, use one of the following characters:

- 0** Zero indicates that the corresponding indicator must be off for the condition to be true.
- 1** One indicates that the corresponding indicator must be on for the condition to be true.
- X** Indicates that the corresponding indicator is not checked.

### statement

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

### Example

This example specifies that if switches 1 and 2 are on and if switch 3 is off (the other switches are not checked), the procedure called MASTER runs.

```
// IF SWITCH-110XXXXX MASTER
```

## SWITCHn (Individual Switches) Condition

The SWITCHn conditional expression determines the setting of the nth switch.

```
// { IF } SWITCHn- $\left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\}$  statement
   { IFT }
   { IFF }
```

59020031-0

- n** This is a number from 1 through 8 indicating which switch is tested.
- 0** The condition is true if the nth switch is off.
- 1** The condition is true if the nth switch is on.

**statement** This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

### Example

This example specifies that if indicator 2 is off, the switch is turned on.

```
// IF SWITCH2-0 SWITCH X1XXXXXX
```

## string1=string2 (Comparing, Equal To) Condition

The equal to (=) conditional expression determines whether one expression equals another expression. This condition is true if string1 is equal to string2.

```
// { IF } { string1=string2 } statement  
   { IFT } { string1/string2 }  
   { IFP }
```

59020032-0

### string1 and string2

These represent two values, such as parameters, character data, or substitution expressions. Each character string is up to 128 characters long. If the string contains blanks, equal signs (=), slashes (/), greater than signs (>), or hyphens (-), you must enclose the string in apostrophes ('). If the string contains apostrophes, such as o'clock, enter two apostrophes; for example: 'o'clock'.

### statement

This specifies a statement to process. See "Statement Portion of IF Conditional Expression" on page 3-43 for more information.

If either of the two compared values has any alphabetic characters or is enclosed by apostrophes ('), the two strings are padded on the right with blanks to a length of 128. They are compared on a character-by-character basis, starting with the character which is farthest left.

If both of the compared values have only numeric characters (and are not enclosed by apostrophes), the two strings are padded on the left with decimal zeros to a length of 128, and they are compared on a character-by-character basis, starting with the character which is farthest left. Numeric digits preceded by a plus (+) or minus (-) sign are considered numeric strings.

For example,

```
// IF ABCD='ABC ' PROCA  
// IF 13=1 PROCB
```

compares as:

```
ABCD:   ABCD  
ABC:    ABC  
13:     000...0013  
1:      000...0001
```

The ... indicates more zeros.

Numeric data is either positive or negative, for example, +25 or -3. The plus (+) sign is unnecessary: for example, 25 is assumed as +25. The value -0 is treated as 0. You must enclose character data preceded by a sign in quotes. For example: '+ABCDE'. If the first character of a string is a plus (+) or minus (-) sign, *all* of the remaining characters in the string must be numeric. If a sign appears in a string in any position but the first position, it is considered as data. You can specify strings in any of the following ways:

```
string1=string2
'string1'=string2
'string1'='string2'
```

**Example 1**

This example specifies that if the third positional parameter is PAYROLL, the procedure named PAYROLL runs.

```
// IF ?3?=PAYROLL PAYROLL
```

**Example 2**

This example specifies that if the first positional parameter does not have a value, the PROMPT OCL statement processes.

```
// IF ?1?/ PROMPT MEMBER-SCREENS,FORMAT-F1
```

**string1>string2 (Comparing, Greater Than) Condition**

The greater than (>) conditional expression determines whether one expression is greater than another expression. This condition is true if string1 is greater than string2.

```
// { IF } string1>string2 statement
   { IFT }
   { IFF }
```

59020033:0

**string1 and string2**

These represent two values, such as parameters, character data, or substitution expressions. Each character string is up to 128 characters long. If the string contains blanks, equal signs (=), slashes (/), greater than signs (>), or hyphens (-), the string must be enclosed in apostrophes ('). For information about how the strings are compared and how you can enter them, see “string1=string2 (Comparing, Equal To) Condition” on page 3-40.

**statement**

This specifies a statement to process. See “Statement Portion of IF Conditional Expression” on page 3-43 for more information.

**Example**

If the value of the fourth positional parameter is greater than 100, the procedure called MASTER runs.

```
// IF ?4?>100 MASTER
```

**VOLID (Diskette and Tape Volume IDs) Condition**

The VOLID conditional expression determines whether the correct diskette or tape is in use. This condition is true if the volume ID of the specified diskette or tape is the same as the specified volume ID.

If the diskette drive or the tape drive is empty, a message is displayed. The operator can then insert a diskette or tape into the specified drive or cancel the job.

```
// { IF } VOLID-{volume id} statement
   { IPT }
   { IFF }
```

59020034-0

**volume id** This specifies the volume ID searched for.

**location** This specifies the location of the diskette or tape that is checked if the system has a diskette drive or tape drive. Diskette locations are allowed for System/36 compatibility, but only syntax checking is performed. The diskette device defined in your System/36 environment configuration is checked. The location can be any of the following:

**S1, S2, or S3**

This identifies an individual diskette slot. If you do not specify a location, S1 is assumed.

**ALLS** This specifies searching of all slots (S1, S2, and S3).

**M1.nn** This identifies a location within magazine 1. **nn** is any number from 01 through 10. For example, M1.04 indicates slot 4 within magazine 1.

**M2.nn** This identifies a location within magazine 2. **nn** is any number from 01 through 10.

**ALL1** This specifies the searching of all diskettes in magazine 1.

**ALL2** This specifies the searching of all diskettes in magazine 2.

**ALL** This specifies searching of all diskettes in both magazines.

**T1** This specifies searching of the tape in tape drive 1.

**T2** This specifies searching of the tape in tape drive 2.

**TC** This specifies searching of the tape in the tape cartridge drive.

**statement** This specifies a statement to process. See "Statement Portion of IF Conditional Expression" on page 3-43 for more information.

### Example 1

This example specifies that if the volume ID of the inserted diskette is not VOL001, the message Insert correct diskette in slot 1 displays.

```
// IFF VOLID-VOL001 PAUSE 'Insert correct diskette in slot 1'
```

### Example 2

This example specifies that if the volume ID of the diskette in location M2.03 is TEST, the information on the diskette is removed.

```
// IF VOLID-'TEST,M2.03' INIT TEST,,DELETE,M2.03
```

### Example 3

This example specifies that if the volume ID of the tape mounted on tape drive 1 is TEST, the information on the tape is removed.

```
// IF VOLID-'TEST,T1' TAPEINIT T1,STDLABEL,TEST,,ERASE
```

## Statement Portion of IF Conditional Expression

The statement portion in an IF expression is any of the following:

- An OCL statement without the // characters that normally precede an OCL statement.
- A statement that calls another procedure.
- A utility control statement without the // characters that normally introduce a utility control statement.
- Another IF, IFT, or IFF expression without the // characters that normally precede the expression. For example:

```
// IF PROC-PAYROLL IF SWITCH3-1 PAYROLL
```

specifies that if a procedure member named PAYROLL is in the system library and if indicator 3 is on, the PAYROLL procedure runs.

- A // \*, // \*\*, CANCEL, EVALUATE, GOTO, PAUSE, RESET, or RETURN statement without the // characters that normally precede the statement.

---

## ELSE Expressions

The ELSE expression is an optional expression that you can only use with the IF expression. Use the ELSE expression when processing a statement because an IF expression was not satisfied. The ELSE expression processes only when the one or more IF expressions preceding the ELSE expression are not satisfied. You can use the ELSE expression with all forms of the IF expression (IF, IFT, and IFF).

The ELSE expression has the following format:

```
// ELSE statement
```

59020035-0

**statement** This is any statement that is valid for the IF conditional expression. See “Statement Portion of IF Conditional Expression.”

Example of IF and ELSE expressions:

```
// IF ?1?= RETURN
// ELSE DELETE ?1?
```

In this example:

- If you do not enter the first parameter of the procedure, the RETURN statement processes.
- If you enter the first parameter on the procedure (parameter 1 is not blank), the DELETE procedure runs, to delete the file specified by the first parameter.

You can only have one ELSE expression after the IF expression and it must be the first expression in that line. If the ELSE expression does not immediately follow the IF expression, the ELSE expression is ignored. An IF expression can follow an ELSE expression in a conditional statement. For example:

```
// IF ?2?= PAYROLL DAILY
// ELSE IF ?2?=YEAREND PAYROLL YEAREND
//     ELSE PAYROLL WEEKLY
```

## // \* (Informational Message) Statement

In this example, if the second parameter is blank, the PAYROLL procedure with the parameter DAILY is called. If the second parameter is YEAREND, it passes to PAYROLL and the PAYROLL procedure runs. If the second parameter is neither blank nor YEAREND, the parameter WEEKLY passes to PAYROLL and PAYROLL runs.

You can also have conditional expressions in which more than one IF expression (in the same line) comes before an ELSE expression, as in the following:

```
// IF SWITCH1-1 IF SWITCH2-1 PROCA
// ELSE PROCB
```

In this case, procedure PROCA only runs when both switch 1 and switch 2 are on. If either switch 1 or switch 2 is off (that is, if either of the tests are false), procedure PROCB runs.

Continuation of a statement following an ELSE expression onto two or more lines will make the continuation expression incorrect. For example:

```
// IF SWITCH1-1 PROCA
//   ELSE FILE NAME-WORK,
//       UNIT-F1
```

results in an error when switch 1 is on because the system processes the two statements:

```
// PROCA
//       UNIT-F1
```

which causes an error.

---

## // \* (Informational Message) Statement

The // \* statement causes an informational message to display at the display station that submitted the job unless the // \* statement is in a job run from the job queue or a job that released its requester display station. In those cases, the // \* statement causes a message to display at the system console.

This type of message does not display if you entered INFOMSG NO previously. See "INFOMSG Control Command" on page 6-12 for more information.

```
// * {message id code}
      {'message text'}
```

59020036-0

### message id code

This specifies the 4-digit message identification code of a message in the current first-level message member. The text of the specified message number displays. See "MEMBER OCL Statement" on page 5-59 for information on assigning a first-level message member. For information about creating message members, see "CREATE Procedure" on page 4-66.

### 'message text'

This specifies the message text that displays. Enclose the message text in apostrophes ('). Use any character in the message text. If the message contains embedded apostrophes (such as the apostrophe in

## // \*\* (System Console Message) Statement

o'clock), then enter each embedded apostrophe as two apostrophes (for example: o'clock).

The message text contains up to 505 characters. The entire message displays, 75 characters at a time. The message lists 75 characters at a time in the history file.

The message text can contain double-byte characters. If you attempt to send double-byte characters to a display station that cannot display double-byte characters, periods replace the double-byte characters.

### Example 1

In the following example, the message indicates that the PAYROLL procedure is running.

```
// * 'PAYROLL Procedure Running'  
// LOAD PAYROLL  
// RUN
```

### Example 2

In the following example, the message indicates that the PAYROLL procedure is running. However, the message displays only when the procedure runs from the keyboard.

```
// IF JOBQ-NO IF EVOKED-NO * 'PAYROLL Procedure Running'  
// LOAD PAYROLL  
// RUN
```

### Example 3

In the following example, the message indicated by number 0005 displays from the first-level message member named MESSAGES.

```
// MEMBER USER1-MESSAGES,LIBRARY-MYLIB  
// * 0005  
// LOAD PAYROLL  
// RUN
```

---

## // \*\* (System Console Message) Statement

The // \*\* statement displays a message on the system operator (QSYSOPR) message queue. The job stops processing until the system operator responds to the message. The message displays with a 0 option response.

```
// ** {message id code}  
      {'message text'}
```

59020037-0

### message id code

This specifies the 4-digit message identification code of a message in the current first-level message member. The text of the specified message displays. See "MEMBER OCL Statement" on page 5-59 for information on assigning a first-level message member. For information about creating message members, see "CREATE Procedure" on page 4-66.

## EVALUATE Statement

### 'message text'

This specifies the message text to display. Enclose the message text in apostrophes ('). You can use any character in the message text. If the message contains embedded apostrophes (such as the apostrophe in o'clock), then enter each embedded apostrophe as two apostrophes (for example: o'clock).

The message text contains up to 504 characters. The entire message displays, 75 characters at a time. Each message has a 0 option response. The message lists 75 characters at a time in the history file.

The message text can contain double-byte characters. If you try to send double-byte characters to a display station that cannot display double-byte characters, periods display for the double-byte characters.

### Example

The following example, contained in a procedure, causes the message Insert the next diskette to appear on the system console display:

```
SAVE FILE1,,,VOL001
// ** 'Insert the next diskette'
SAVE FILE2,,,VOL002
```

---

## CANCEL Statement

The CANCEL statement cancels a procedure and returns control to the keyboard for the next statement. If you use a CANCEL statement to cancel a MRT procedure, an error message is issued to all active requesters of that procedure.

When a job is ended with the CANCEL statement, it is considered an abnormal ending. See the *System/36 Environment Programming* book for more information on abnormal ending of jobs. See the description of the RESET statement for an example of ending a job normally.

```
// CANCEL
```

59020038-0

### Example

This example uses the CANCEL statement to cancel a procedure when an IF expression is not satisfied:

```
// IFF DATAF1-FILEA CANCEL
```

---

## EVALUATE Statement

The EVALUATE statement allows you to do the following:

- Assign values to a parameter.
- Perform addition, subtraction, multiplication, and division of values.
- Evaluate substitution expressions.
- Set the return code.

You can evaluate more than one item at a time.

The EVALUATE statement can contain any type of data. For example, it can contain arithmetic expressions, substitution expressions, or comments.

You can use the EVALUATE statement as the result of a series of conditional tests. It makes your conditions easier to follow. For example:

```
// IF DATAF1-FILEA IF SWITCH1-1 EVALUATE (EVERYTHING IS FINE)
// ELSE CANCEL
PROCA
```

If the tests are true, the EVALUATE and the PROCA statements process. Otherwise, the CANCEL statement processes. Note the comment EVERYTHING IS FINE on the EVALUATE statement. Anything not recognized as an expression is considered a comment.

## Assigning Values

You can assign more than one parameter on the EVALUATE statement, but you must separate each parameter assigned by one or more blanks. The expressions are evaluated from left to right. You can use the results of one expression in a following expression.

```
// EVALUATE Pn [ ,length ] =expression
```

59020039.0

**P** This indicates a parameter is assigned.

**n** This indicates the number of the procedure parameter (1 through 64) into which the expression is saved. **n** is either a number or a substitution expression with a numeric value.

**length** This is an optional value that specifies the length (number of characters) of the resulting expression. The length is a number from 1 through 15. If you do not specify the length, the length of the expression is used. Any leading zeros are not used in the final result. Specify the length only for numeric values. If a length is specified and the number of significant digits in the result is greater than the specified length, the result is truncated on the left. For example:

```
P3,2=12345
```

causes parameter 3 to set to '45'.

If you specify a length and the number of significant digits in the result is smaller than the specified length, the result is padded on the left with zeros. For example:

```
P3,5=123
```

causes parameter 3 to set to '00123'.

If you do not specify length, the length of the expression is used. For example:

```
P3=876253
```

causes parameter 3 to set to '876253'.

For negative numbers, reserve a place for the minus sign (-). For example:

## EVALUATE Statement

```
// EVALUATE P5,2=-12-1
```

parameter 5 contains 13 instead of -13 because there was no room allowed for the minus sign.

### expression

This indicates an expression. The expression is an integer constant, a substitution expression having the value of an integer, a substitution expression having a value from a statement in a message member, or a character string. The character string can contain IGC characters. If the character string contains any embedded blanks, enclose it in apostrophes ('). For example: P5='ab c'. To set a parameter off, enter 2 consecutive apostrophes ("). For example: P5="".

### Example 1

This example assigns the value 123 to parameter 1, the value 456 to parameter 2, and sets parameter 3 off. Both parameters 1 and 2 have a length of 3. Parameter 3 has a length of 0.

```
// EVALUATE P1=123 P2=00456 P3=''
```

### Example 2

This example assigns the value 00123 to parameter 1. Parameter 1 has a length of 5.

```
// EVALUATE P1,5=123
```

### Example 3

This example assigns the value 'EXAMPLE DATA' to parameter 1. Parameter 1 has a length of 12.

```
// EVALUATE P1='EXAMPLE DATA'
```

### Example 4

If the operator has a user ID of MAT, this example assigns the value MAT to parameter 1. Parameter 1 has a length of 3.

```
// EVALUATE P1=?USER?
```

### Example 5

If parameter 3 has a value of 123, parameter 9 has a value of 30, and parameter 10 has a value of 5, this example assigns the value 00123 to parameter 30. Parameter 30 has a length of 5.

```
// EVALUATE P?9?,?10?=??3?
```

is evaluated as:

```
// EVALUATE P30,5=123
```

## Adding, Subtracting, Multiplying, or Dividing

You can assign more than one parameter on the EVALUATE statement, but you must separate each parameter assigned by one or more blanks. The parameters assigned are evaluated from left to right.

```
// EVALUATE Pn [ ,length ] =expression
```

59020040-0

- P** This indicates a parameter is assigned.
- n** This indicates the number of the procedure parameter (1 through 64) into which the expression is saved. **n** is either a number or a substitution expression with a numeric value.
- length** This is an optional value specifying the length (number of characters) of the resulting expression. The length is a number from 1 through 15. If you do not specify the length, the length of the expression is used. The final result does not use any leading zeros. For more information about the length parameter, see “Assigning Values” on page 3-47.

#### expression

This indicates one or a combination of the standard arithmetic operations of addition, subtraction, multiplication, or division. These expressions are made up of numbers (integer constants or a substitution expression having the value of an integer), signs (+, -, \* for multiplication, and / for division), and, optionally, left and right parentheses to group operations. Characteristics of such arithmetic expressions, and the rules for constructing them, are as follows:

The minus (-) sign can indicate a negative number as well as the subtraction operation. For example:

```
// EVALUATE P1=-4+-5
```

is valid and assigns -9 to parameter 1.

Blanks are not allowed within one assignment expression, for example:

```
// EVALUATE P1 = 1 + 2
```

is not valid, but:

```
// EVALUATE P1=1+2
```

is valid.

Multiplication and division operations have a higher priority than addition and subtraction operations. In other words, multiplication and division are performed before addition and subtraction in an expression.

A left parenthesis indicates the start of an algebraic operation, and a right parenthesis indicates the end of an algebraic operation. Operations grouped by parentheses are performed before operations not grouped by parentheses, regardless of their relative priorities.

When division is performed, remainders are dropped. No rounding is performed. To obtain the remainder, multiply the quotient by the original divisor, and then subtract that result from the dividend. For example:

```
// EVALUATE P1=17/5 P2=17-(?1?*5)
```

assigns the quotient of the division operation to parameter 1 and the remainder to parameter 2.

If you attempt to divide by zero, an error message issues.

## EVALUATE Statement

The result and each operand of the expression are limited to 15 digits. If you attempt evaluation of an expression with a result greater than 15 digits, the results are unpredictable.

### Example 1

This example assigns the value 246 to parameter 1 and the value -1 to parameter 2. Parameter 1 has a length of 3 and parameter 2 has a length of 2.

```
// EVALUATE P1=123+123 P2=0001+2-4
```

### Example 2

This example assigns the value 00246 to parameter 1. Parameter 1 has a length of 5.

```
// EVALUATE P1,5=123+123
```

### Example 3

If parameter 3 has a value of 123, parameter 9 has a value of 30, and parameter 10 has a value of 5, this example assigns the value 00125 to parameter 30. Parameter 30 has a length of 5. For example,

```
// EVALUATE P?9?,?10?=?3?+2
```

is evaluated as:

```
// EVALUATE P30,5=123+2
```

### Example 4

If parameter 1 has a value of 3 and parameter 2 has a value of 7, this example assigns the value 25 to parameter 3. Because multiplication and division operations have a higher priority than addition and subtraction operations, the value in parameter 1 is multiplied by the value in parameter 2 first. Then 4 is added to the result.

```
// EVALUATE P3=4+?1?*?2?
```

### Example 5

If parameter 1 has a value of 5, this example assigns the value 30 to parameter 2. Because it is grouped by parentheses, the addition operation is performed before the multiplication operation.

```
// EVALUATE P2=3*(?1?+5)
```

### Example 6

This example shows how to create a file based upon the size of two work files. Files CUSTMST and WORK are two resident files that contain several records.

```
// EVALUATE P64=?F'A,CUSTMST'?+?F'A,WORK'?  
BLDFILE NEWFILE,S,RECORDS,?64?,100
```

For example, if file CUSTMST contained 200 records and file WORK contained 150 records, the statements evaluate as:

```
// EVALUATE P64=200+150  
BLDFILE NEWFILE,S,RECORDS,350,100
```

## Evaluating Substitution Expressions

```
// EVALUATE expression
```

59020041-0

### expression

This is one or more substitution expressions. You must separate each expression by one or more blanks. The expressions evaluate from left to right. You can use the results of one expression in a following expression.

### Example 1

This example assigns the value ABC to parameter 1 only if parameter 1 does not have a value.

```
// IF ?1?/ EVALUATE ?1'ABC'?
```

### Example 2

This example assigns the value MYLIB to parameter 3 if parameter 3 does not have a value and assigns the value PROC to parameter 2 if parameter 2 does not have a value.

```
// EVALUATE ?3'MYLIB'? ?2'PROC'?
```

## Setting the Return Code

You can save the return code from one job, run a second job, and then set the return code value back to the value it was for the first job.

```
// EVALUATE CD=nnnn
```

59020042-0

**CD** This indicates that the return code is restored.

**nnnn** This indicates the 4-digit return code value to restore.

### Example

In this example, the first EVALUATE statement saves the return code value for program 1, and the second EVALUATE statement restores the return code value for program 3.

```
// LOAD PROG1
// RUN
* Save return code for PROG3
// EVALUATE P64,4=?CD?
// ALLOCATE UNIT-I1,WAIT-NO
* This will change return code if diskette drive is unavailable
// LOAD PROG2
// RUN
* Restore return code from PROG1
// EVALUATE CD=?64?
// LOAD PROG3
// RUN
```

---

### GOTO and TAG Statements

The GOTO and TAG statements allow you to branch around groups of statements within a procedure. By using GOTO and TAG statements instead of several IF expressions, you can reduce the number of expressions that the System/36 environment processes during procedure generation.

```
// GOTO label
```

```
// TAG label
```

59020043-0

A GOTO statement causes the System/36 environment to branch to the statement following a TAG statement with the same label. (More than one TAG statement in a procedure can have the same label. The system branches to the next TAG statement with the required label.) The System/36 environment searches for the target TAG statement within the procedure. The search begins with the statement following the GOTO statement. If the System/36 environment does not find the target statement before it reaches the end of the procedure, the System/36 environment goes to the first statement in the procedure and resumes the search.

The following is an example of GOTO and TAG statements:

```
* Example of GOTO and TAG statements
// GOTO A
.
.   These statements are skipped
.
// TAG A   Processing resumes after this statement
.
.
.
```

You cannot substitute the word TAG. For example, the following is not valid:

```
// EVALUATE P1=TAG
// GOTO A
.
.
// ?1? A
```

#### Example

A procedure named PROC runs four procedures: STEP1, STEP2, STEP3, and STEP4. Normally, to start PROC, the operator enters:

```
PROC
```

If the operator cancels PROC at some point, you can restart the procedure at the canceled step by entering:

```
PROC RESTART,step name
```

**step name** This is STEP1, STEP2, STEP3, or STEP4.

The following are the statements in PROC:

```

// IF ?1?= IF ??= GOTO STEP1      (STEP1 IS THE DEFAULT)
// IFF ?1?=RESTART * 'INCORRECT ENTRY'
// IFF ?1?=RESTART CANCEL
// IF ??=STEP1 GOTO STEP1
// IF ??=STEP2 GOTO STEP2
// IF ??=STEP3 GOTO STEP3
// IF ??=STEP4 GOTO STEP4
// * 'INCORRECT ENTRY'
// CANCEL
// TAG STEP1
STEP1
// TAG STEP2
STEP2
// TAG STEP3
STEP3
// TAG STEP4
STEP4

```

---

## PAUSE Statement

The PAUSE statement stops the processing of a job and displays a message at the display station that submitted the job. If the PAUSE statement is in a job run from the job queue or a job that released its requester display station, the PAUSE statement displays a message at the system console. The operator can then restart the job by selecting option 0. The system then continues processing the statements that follow the PAUSE statement.

Although you can use up to 500 characters in the message, only the first 68 characters display.

```

// PAUSE      ['message text']

```

59020044-0

### 'message text'

This specifies the message to display when the job pauses. You must contain the message within single apostrophes ('). If the message text contains embedded apostrophes (such as the apostrophe in o'clock), then enter the embedded apostrophes as two apostrophes (for example: 'o'clock').

The message text can contain double-byte characters. If there is an attempt made to run a procedure containing a double-byte message following a PAUSE statement at a display station that cannot display double-byte characters, the double-byte characters are replaced by periods.

If you do not specify message text, the following message displays:

```
PAUSE -- when ready, enter 0 to continue
```

## RESET Statement

### Example

In this example, the PAUSE statement informs the operator that the payroll program is complete. The operator enters a 0 (zero) to continue.

```
// LOAD PAYROLL
// FILE NAME-EMPLOY
// RUN
// PAUSE 'Payroll program is complete'
```

---

## RESET Statement

You can have up to 255 procedure levels. However, for some applications, you may need many more levels of procedure calls. For example, a procedure might call itself an undetermined number of times. The RESET statement calls a procedure and specifies that it is treated as a first-level procedure. That is, when the procedure called by the RESET statement completes, control does not return to the procedure level containing the RESET statement.

```
// RESET  procedure name [ ,library name ] [ parm1,parm2,...
                                     program data
                                     *ALL ]
```

59020045-0

### procedure name

This specifies the name of the procedure called.

### library name

This specifies the name of the library containing the procedure.

### parm1,parm2,...

These are the optional procedure parameters. The procedure specified by procedure name becomes a first-level procedure. (Control does not return to the procedure containing the RESET statement.) Specify a maximum of 64 parameters. See "INCLUDE OCL Statement" on page 5-49 for more information about how to specify parameters.

### program data

This is the data to pass to a program on its first input operation from the display station.

**\*ALL** This specifies passing of all 64 parameters from the current procedure level to the specified procedure.

### Example 1

A procedure called PROCA displays a list of options to an operator and tells the operator to enter one of the options. PROCA then calls the procedure specified by the selected option. When that procedure completes, PROCA calls itself and the process repeats. In this application, PROCA must be able to call itself any number of times. Following are the statements in the procedure member called PROCA.

```

* THIS IS PROCEDURE PROCA
// * 'ENTER ONE OF THE FOLLOWING OPTIONS:'
// * ' 1. RUN WEEKLY PAYROLL'
// * ' 2. PRINT CURRENT INVENTORY'
// * ' 3. RUN WEEKLY BILLING'
// * ' 4. STOP'
// IF ?1R?=4 CANCEL
// IF ?1?=1 PAYROLL WEEKLY
// IF ?1?=1 RESET PROCA
// IF ?1?=2 INVENT
// IF ?1?=2 RESET PROCA
// IF ?1?=3 BILLING WEEKLY
// IF ?1?=3 RESET PROCA
// * '?1? IS A WRONG OPTION'
// RESET PROCA

```

### Example 2

This example shows how the RESET statement can be used to end a System/36 job. The job ends normally by doing a RETURN from the first-level procedure, regardless of how many procedure levels are active. Procedure QUIT contains a RETURN statement. Procedure PROCB contains a test to determine if the job should end. If the job should end, the QUIT procedure is called as the first-level procedure and the job ends normally.

```

Procedure QUIT:
// RETURN

```

```

Procedure PROCB:
// LOAD PGMA
// RUN
// IF SWITCH 1-1 RESET QUIT
// LOAD PGMB
// RUN

```

---

## RETURN Statement

If the RETURN statement processes in a first-level procedure, the procedure returns to the requesting display station for the next statement if there is further input required. If further input is required and a requesting display station does not exist (for example, the procedure runs from the job queue), an error message displays. If further input is not required, the job terminates. If it processes in a nested procedure, the RETURN statement returns to the calling procedure for the next statement.

```

// RETURN [ *ALL ]

```

59020046-0

If you specify \*ALL, all 64 procedure parameters from the current procedure level pass to the previous procedure level. This causes complete replacement of all the parameters for the previous procedure.

If you do not specify \*ALL, the parameters for the previous procedure level remain unchanged.

## RETURN Statement

### Example 1

This example shows how to end a procedure using a RETURN statement so that the system does not waste time processing comment statements.

```
// LOAD PROGA
// FILE NAME-INPUT
// FILE NAME-OUTPUT
// RUN
// RETURN
* This procedure reads a file named INPUT, and
* creates a file named OUTPUT.
```

### Example 2

This example passes parameters from procedure PROC2 back to the procedure that called PROC2.

```
* Procedure PROC2
// EVALUATE P1=FILEA P2=INPUT P3=OUTPUT
// RETURN *ALL
```

---

## Chapter 4. Procedures

This chapter describes the IBM-supplied OS/400 System/36 environment procedures.

The following information is given for each procedure described in this chapter:

- The function of the procedure.
- The syntax format of the procedure command that calls the procedure. For a description of the rules used for syntax formats, see “Conventions Used for Describing Syntax Formats” on page 1-3.
- Descriptions of the parameters used in the procedure command.
- One or more examples of how to use the procedure and how to enter the procedure command or statements to run the procedure.

When you are signed on to the System/36 environment, you can use the system help support to run these procedures. The system help support is made up of menus, prompt displays, and help text. The menus allow you to select a task you want to perform. When you select an item from a menu, either:

- Another menu appears (as you select options, each one gets more specific about the task you want to perform).
- A prompt display for a procedure or command appears.

The prompt displays allow you to run procedures to do the task. The help text explains the menus, the menu options, the procedures and commands, and the parameters for the procedures and commands.

For more information about the system help support, see “HELP Procedure” on page 4-109.

Table 4-1 shows the name of the utility program or the name of the OCL statement that is run or processed by the System/36 environment procedures that use utilities or OCL statements.

*Table 4-1 (Page 1 of 2). System/36 Environment Procedures*

<b>System/36 Environment Procedure</b>	<b>System/36 Environment Utility Run</b>	<b>OCL Statement Processed</b>
BLDFILE	\$FBLD	
BLDINDEX	\$FBLD	
BLDGRAPH	\$DPGR	
BLDLIBR	\$MAINT	
BLDMENU	\$BMENU	
	\$MAINT	
	\$MGBLD	
	\$SFGR	
CATALOG	\$LABEL	
CHNGEMEM	\$MAINT	
COPYDATA	\$COPY	
COPY11	\$DUPRD	
COPYPRT	\$UASF	

Table 4-1 (Page 2 of 2). System/36 Environment Procedures

System/36 Environment Procedure	System/36 Environment Utility Run	OCL Statement Processed
CREATE	\$UASC	
DATE	\$MGBLD	DATE
DELETE	\$DELET	
FLIB		FILELIB
ERR	\$CPPE	
FORMAT	\$SFGR	
FROMLIBR	\$MAINT	
HELP	\$HELP	
INIT	\$INIT	
JOBSTR	\$MAINT	JOBQ
KEYSORT	\$DDST	
LIBRLIBR	\$MAINT	
LINES		FORMS
LISTDATA	\$COPY	
LISTFILE	\$BICR	
	\$COPY	
	\$MAINT	
	\$TCOPY	
LISTLIBR	\$MAINT	
LOG		LOG
NOHALT		NOHALT
PASSWORD	\$PRPWD	
PRINT		FORMS
PRINTKEY	\$SETCF	
PRTGRAPH	\$DPGP	
REMOVE	\$MAINT	
RENAME	\$RENAM	
RESPONSE	\$ARSP	
RESTLIBR	\$MAINT	
RESTORE	\$COPY	
SAVE	\$COPY	
SAVELIBR	\$MAINT	
SET	\$SETCF	
SLIB		LIBRARY
SORT	#GSORT	
SRTX	#KASRT	
SWITCH		SWITCH
SYSLIST		SYSLIST
TAPECOPY	\$TCOPY	
TAPEINIT	\$TINIT	
TOLIBR	\$MAINT	
TRANSFER	\$BICR	

## Managing Licensed Programs

To help you manage the licensed programs on your AS/400 system, use the Manage Licensed Programs (LICPGM) menu. This menu provides options to install, delete, save, and display licensed programs and national languages.

If you want to install, save, or delete a licensed program from an OCL procedure or CL program, you can use the following commands:

- To do a similar function to a System/36 LOAD procedure, use the Restore Licensed Program (RSTLICPGM) command.
- To do a similar function to a System/36 SAVE procedure, use the Save Licensed Program (SAVLICPGM) command.
- To do a similar function to a System/36 DROP procedure, use the Delete Licensed Program (DLTLICPGM) command.

See the *CL Reference* book for more information on the RSTLICPGM, SAVLICPGM, and DLTLICPGM commands. See the *Software Installation* book for more information on managing licensed programs on your system.

---

## #STRTUP1 Procedure

You can create the #STRTUP1 procedure to start your own jobs that need to be run as part of the initial program load (IPL) process *while* other jobs are run. #STRTUP1 is a reserved procedure name that is searched for during IPL. If the procedure is found, it begins running. If it is not found, no error occurs. Other jobs can be started while this job runs, and other operators are allowed to sign on. No parameters can be passed to the procedure.

You can use this procedure to do tasks you want done each time IPL is performed, without keeping other jobs from starting while this job runs.

**Note:** The #STRTUP1 procedure and the #STRTUP2 procedure will both run after the program starts, while other users can be signing on.

You can place the #STRTUP1 procedure in the system library (#LIBRARY) or in the system operator's sign-on library.

To make your own procedures, see Chapter 2, "Making Your Own Procedures."

### Example

This example shows a procedure you could include in the #STRTUP1 procedure and store in the system library (#LIBRARY).

```
* #STRTUP1 procedure
* Set the system default automatic response severity level to 3
NOHALT 3,SYSTEM
```

---

## #STRTUP2 Procedure

You can create the #STRTUP2 procedure to start your own jobs that need to be run as part of the initial program load (IPL) process *while* other jobs are run. #STRTUP2 is a reserved procedure name that is searched for during IPL. If the procedure is found, it begins running. If it is not found, no error occurs. Other jobs can be started while this job runs, and other operators are allowed to sign on. No parameters can be passed to the procedure.

You can use this procedure to do tasks you want done each time IPL is performed, without keeping other jobs from starting while this job runs.

You can place the #STRTUP2 procedure in the system library (#LIBRARY) or in the system operator's sign-on library.

**Example**

This example shows a procedure you could include in the #STRTUP2 procedure and store in the system library (#LIBRARY).

```
* #STRTUP2 procedure
* List the contents of the disk
CATALOG
```

---

**ALERT Procedure**

Not supported. On System/36, the ALERT procedure started the alert support, which allowed you to specify which system messages in a predefined subset of system messages should generate alert messages. The ALERT procedure also provided the interface for sending a notification alert. A notification alert allowed you to generate your own message and send it to the remote location designated to receive alerts. Alert messages were sent to the designated location of a communications network through the advanced program-to-program communications (APPC) subsystem.

On the AS/400 system, alerts are generated as a result of an alertable message issued to the local system operator (QSYSOPR) message queue. Messages issued to the System/36 environment program messages display are not alertable.

To do a similar function, use the Change Message Description (CHGMSGD) command, specifying the alert options (ALROPT) parameter for each message that you want to generate an alert message. Use the Send Program Message (SNDPGMMSG) command to send operator or application generated alerts to the system operator message queue (QSYSOPR). Operator generated alerts can also be generated interactively using the Analyze Problem (ANZPRB) command.

See the *CL Reference* book for more information on the CHGMSGD, SNDPGMMSG, and ANZPRB commands. See the *Alerts Support* book for more information on specifying and generating alert messages. See the *System/36 Environment Programming* book for more information on messages in the System/36 environment.

---

**ALOCFLDR Procedure**

Not supported. On System/36, the ALOCFLDR procedure allowed you to reorganize a folder and change its size.

To do a similar function, use the Reorganize Document Library Object (RGZDLO) command.

See the *System/36 Environment Programming* book for more information on using folders in the OS/400 System/36 environment, and the *CL Reference* book for more information on the RGZDLO command. Also, see the *Planning for and Setting Up OfficeVision/400* book for more information on folder management for the AS/400 system.

---

## ALOCLIBR Procedure

The ALOCLIBR procedure is supported as a no-operation command. Only syntax checking of the procedure parameters is done.

On System/36, the ALOCLIBR procedure allowed you to increase or decrease the size of a user library or the system library (#LIBRARY). You could also change the size of a user library's directory.

On the AS/400 system, this function is not required.

See the *System/36 Environment Programming* book for more information on managing your libraries.

<pre> ALOCLIBR [ library name            current library ], [ library size ], [ directory size ], [ A1   A2   A3   A4   block number ], [ INCR   DECR ] </pre>
--

59020048-0

---

## ALTERBSC Procedure

The IBM System/34 ALTERBSC procedure is not supported. On System/36, the user was referred to the ALTERCOM procedure.

To do a similar function, use the Work with Line Descriptions (WRKLIND) command. You can also change a BSC line description by using the Change Line Description (BSC) (CHGLINBSC) command.

See the *CL Reference* book for more information on the WRKLIND and CHGLINBSC commands. See the *System/36 Environment Programming* book for more information on the AS/400 communications configuration.

---

## ALTERCOM Procedure

Not supported. On System/36, the ALTERCOM procedure changed communications items for a specified communications line. Changes only applied to programs run from the same display station as the ALTERCOM procedure.

To do a similar function, use the Work with Line Descriptions (WRKLIND), Work with Device Descriptions (WRKDEV), and Work with Controller Descriptions (WRKCTLD) commands. From the displays these commands show, you can change or create the associated descriptions. You can use the Work with Configuration Status (WRKCFGSTS) command to determine the status of the line or device you want to change.

On the AS/400 system, changes made for a communications line are changed for the system and not just the display station where the changes were made.

On the AS/400 system, not all line characteristics can be changed. If an application requires a line with multiple variations of these characteristics, create multiple line descriptions and vary on the line description needed.

The following is a mapping of the System/36 ALTERCOM procedure parameters to the equivalent OS/400 command parameters:

### **line number**

On System/36, specified the number of the communications line that was to have one or more of its characteristics changed.

On the AS/400 system, specify the line description (LIND) parameter on the appropriate command for changing the line description.

### **SWITCHED**

On System/36, specified that the line was a point-to-point, switched communications line.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

### **AA**

On System/36, specified that if the communications modem was in automatic answer mode, the system was to automatically answer the call.

On the AS/400 system, specify \*YES for the auto answer (AUTOANS) parameter on the appropriate command for changing the line description.

### **MA**

On System/36, specified that the system operator was to manually answer the call.

On the AS/400 system, specify \*NO for the auto answer (AUTOANS) parameter on the appropriate command for changing the line description.

### **MC**

On System/36, specified that the system operator was to manually call the device to be communicated with.

On the AS/400 system, specify \*NO for the auto dial (AUTODIAL) parameter on the appropriate command for changing the line description.

### **remote id**

On System/36, specified the hexadecimal value of the remote station ID.

On the AS/400 system, specify the remote identifier (RMTID) parameter on the Change Controller Description (BSC) (CHGCTLBSC) command.

### **local id**

On System/36, specified the hexadecimal value of the local station ID.

On the AS/400 system, specify the local identifier (LCLID) parameter on the Change Controller Description (BSC) (CHGCTLBSC) command.

### **NONSWTCH**

On System/36, specified that the line was to be a point-to-point non-switched communications line.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

**MULTTRIB**

On System/36, specified that the system was a multipoint tributary station.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

**tributary address**

On System/36, specified the address of the system on the multipoint communications line. The tributary address was the hexadecimal value of one of the pair of tributary station addressing characters.

On the AS/400 system, specify the station address (STNADR) parameter on the appropriate command for changing the line description.

**MULTCONT**

On System/36, specified that the system was a multipoint control station. This parameter cannot be used with BSC.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

**SHM**

On System/36, specified that the line used X.21 short hold mode.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

**NONE**

On System/36, specified that neither blank compression nor truncation was to be performed.

On the AS/400 system, specify \*NO for both the compress and decompress data (DTACPR) parameter and the truncate trailing blanks (TRUNC) parameter of the Change Device Description (BSC) (CHGDEVBSC) command.

**COMPRESS**

On System/36, specified that embedded blanks were to be compressed.

On the AS/400 system, specify \*YES for the compress and decompress data (DTACPR) parameter of the Change Device Description (BSC) (CHGDEVBSC) command.

**TRUNCATE**

On System/36, specified that trailing blanks were to be truncated.

On the AS/400 system, specify \*YES for the truncate trailing blanks (TRUNC) parameter of the Change Device Description (BSC) (CHGDEVBSC) command.

**bsc wait time**

On System/36, specified the number of seconds that BSC was to wait for the System/36 user program to issue a BSC request before it indicated that an error has occurred.

On the AS/400 system, a similar function is provided when specifying the number of retries for transmitting or for receiving temporary text delay (TTD) or wait before transmitting (WACK) before indicating the error and making the line inoperable. Specify the transmit TTD or WACK retry (TMTRTY) and the receive TTD or WACK retry (RCVRTY)

parameters on the Change Line Description (BSC) (CHGLINBSC) command. The default timer values should be used to provide equivalent function to System/36. For TMTRTY, the default timer (CONTTMR parameter) value is two seconds per retry. For RCVRTY, the default timer (RCVTMR parameter) value is three seconds per retry. The System/36 value 999 is the same as the OS/400 \*NOMAX value for the TMTRTY and RCVRTY parameters.

#### **MULTFILE**

On System/36, specified that more than one (3740-type) file could be transmitted or received.

On the AS/400 system, specify \*DEV3740 for the group separator type (GRPSEP) parameter of the Change Device Description (BSC) (CHGDEVBSC) command.

#### **NONMULT**

On System/36, specified that only single 3740-type files could be transmitted or received.

On the AS/400 system, you cannot specify that only single 3740-type files can be transmitted or received. If you specify \*DEV3740 for the group separator type (GRPSEP) parameter of the Change Device Description BSCEL (CHGDEVBSC) command, you cannot control whether a single or multiple 3740-type files can be transmitted or received.

#### **record separator**

On System/36, specified the character to be used to indicate the end of one record and the beginning of another when multiple records per block were sent.

On the AS/400 system, specify \*SEP for the blocking type (BLOCK) parameter and specify the separator character (SEPCHAR) parameters of the Change Device Description (BSC) (CHGDEVBSC) command.

#### **FULL**

On System/36, specified that the full rated speed of the modem was to be used.

On the AS/400 system, specify \*FULL on the modem data rate select (MODEMRATE) parameter of the appropriate command for changing the line description.

#### **HALF**

On System/36, specified that half of the full rated speed of the modem was to be used. This was only valid if the modem supports half rate.

On the AS/400 system, specify \*HALF on the modem data rate select (MODEMRATE) parameter of the appropriate command for changing the line description.

#### **SNBU**

On System/36, specified that the switched network backup line was to be used if a failure occurred on the nonswitched primary line.

This was only valid if the modem and the remote system support switched network backup.

If SNBU was specified for BSC, and manual call, manual answer, or automatic answer was not specified, the connection either became manual call or manual answer, depending on the first communications line operation done by the program. If the first line operation was a

transmit operation, manual call was assumed. If the first line operation was a receive operation, manual answer was assumed.

On the AS/400 system, SNBU must be defined for the line and/or controller when it is created. However, it may be activated or deactivated by specifying the active switched network backup (ACTSNBU) parameter of the appropriate command for changing the line description or changing the controller description.

**NOSNBU** On System/36, specified that no switched network backup line was to be used.

On the AS/400 system, SNBU must be defined for the line and/or controller when it is created. However, it may be activated or deactivated by specifying the active switched network backup (ACTSNBU) parameter of the appropriate command for changing the line description or changing the controller description.

#### **bsc retry count**

On System/36, specified the number of times a transmission was to be attempted if an error occurs.

On the AS/400 system, specify the contention state retry (CTNRTY) parameter and the data state retry (DTASTTRTY) parameter of the Change Line Description (BSC) (CHGLINBSC) command.

#### **primary sdlc time-out value**

On System/36, specified the primary SDLC time-out value in half-second increments.

On the AS/400 system, specify the idle timer (IDLTMR) parameter of the Change Line Description (SDLC) (CHGLINSDLC) command. Other timer values can also be specified on the CHGLINSDLC command.

#### **sdlc retry count**

On System/36, specified the number of retries of a transmission that were to be attempted.

On the AS/400 system, specify the frame retry count (FRAMERTY) parameter of the Change Line Description (SDLC) (CHGLINSDLC) command. Other retry values can also be specified on the CHGLINSDLC command.

#### **secondary sdlc inactivity time-out value**

On System/36, specified the time period that determines when the primary system was to be considered no longer active for a secondary SDLC nonswitched line.

On the AS/400 system, specify the inactivity timer (INACTTMR) parameter of the Change Line Description (SDLC) (CHGLINSDLC) command. Other timer values can also be specified on the CHGLINSDLC command.

See the *CL Reference* book for more information on the CHGLINSDLC, CHGCTLBSC, CHGLINBSC, CHGDEVBSC, WRKLIND, WRKDEVD, WRKCTLD, and WRKCFGSTS commands. See the and *System/36 Environment Programming* book for more information on the AS/400 communications configuration.

---

## ALTERSDL Procedure

The IBM System/34 ALTERSDL procedure is not supported. On System/36, the user was referred to the ALTERCOM procedure.

To do similar function, use the Work with Line Descriptions (WRKLIND) command. You can also change an SDLC line description by using the Change Line Description (SDLC) (CHGLNSDLC) command.

See the *CL Reference* book for more information on the WRKLIND and CHGLNSDLC commands. See the *System/36 Environment Programming* book for more information on the AS/400 communications configuration.

---

## APAR Procedure

Not supported. On System/36, the APAR procedure created diskette or tape files that contained information that could help IBM service personnel find and correct programming problems that might have occurred in the system.

To do similar function, use the Create APAR (CRTAPAR) command.

See the *CL Reference* book for more information on the CRTAPAR command. See the *System Operation* book for more information on working with system problems.

---

## APPNINFO Procedure

Not supported. On System/36, the APPNINFO procedure provided the user access to information needed for advanced peer-to-peer networking (APPN) network problem determination. The types of data available to the user were session-related data, the network configuration information, and the directory of locations.

To do a similar function, use the Display APPN Information (DSPAPPNINF) command. Specify the information type (INFTYPE) as follows:

- \***TOPOLOGY** To access the network configuration (topology) information
- \***LCLNODE** To access the directory information
- \***SSN** To access the session information

On the System/36, the endpoint session information provided by APPNINFO included a PIU wrap and session error data. On the AS/400 system, this information is not provided. The DSPAPPNINF command can only process input and output files located on the system. DSPAPPNINF does not access other storage devices.

See the *CL Reference* book and the *APPC Programming* book for more information on the DSPAPPNINF command. See the *APPC Programming* book for more information on using APPN.

---

## ARCHIVE Procedure

Not supported. On System/36, the ARCHIVE procedure copied a folder member or all marked folder members onto disk, diskette, tape, or tape cartridge. The ARCHIVE procedure could leave the member in the folder, remove the text from the folder and leave only the directory, or remove the entire member from the folder after the member was archived.

To do similar function, use the Save Document Library Object (SAVDLO) command with either the DLO(name) and FLR(name) parameters or the OBJ(\*SEARCH) and CHKFORMRK(\*YES) parameters specified.

See the *CL Reference* book for more information on the SAVDLO command. See the *System/36 Environment Programming* book for more information on using folders in the OS/400 System/36 environment. See the *Planning for and Setting Up OfficeVision/400* book for more information on folder management on the AS/400 system.

---

## ASM Procedure

Not supported. On System/36, the ASM procedure compiled or assembled an Assembler source program or subroutine. You then needed to link-edit the compiled program to produce a program that you could run.

On the AS/400 system, you must convert Assembler programs to a high-level language.

---

## ASMLOAD Procedure

Not supported. On System/36, the ASMLOAD procedure installed the Assembler support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## ASMSAVE Procedure

Not supported. On System/36, the ASMSAVE procedure saved the Assembler support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## AUTO Procedure

This procedure is supported only for compatibility with the IBM System/34. Use the AUTOC procedure. For more information, see "AUTOC Procedure."

---

## AUTOC Procedure

The AUTOC procedure compiles an RPG II program that contains auto report specifications.

For information on auto report, see the *System/36-Compatible RPG II User's Guide and Reference* book.

```

AUTO  source member name, [ source member library ], [ COMP ], [ PRINT ],
      [ current library ], [ NOCOMP ], [ NOPRINT ],
      [ NOXREF ], [ mrt maximum ], [ NONEP ], [ output library ],
      [ XREF ], [ 0 ], [ NEP ], [ source member library ],
      [ SOURCE ], [ DEBUG ], [ program size ], [ NOHALT ], [ REPLACE ], [ LINK ],
      [ PSOURCE ], [ NODEBUG ], [ NOHALT ], [ HALT ], [ NOREPLAC ], [ NOLINK ],
      [ NOSOURCE ],
      [ NOOBJECT ], [ subroutine library ], [ GEN ], [ work file size ],
      [ OBJECT ], [ source member library ], [ NOGEN ], [ 40 ],
      [ data dictionary name ], [ NOMRO ],
      [ MRO ]

```

59020055-1

**source member name**

This specifies the library source member that contains the auto report program specifications.

**source member library**

This specifies the name of the library that contains the source member to compile. If you do not specify this parameter, the current library is assumed.

**COMP** This specifies that the RPG compiler runs as part of the auto report function. If you do not specify a parameter, COMP is assumed.

**NOCOMP** This specifies that the RPG compiler does not run as part of the auto report function.

**PRINT** This specifies that the compiler listing created by the AUTO procedure is printed. If you do not specify a parameter, PRINT is assumed.

**NOPRINT** This specifies that no listing prints or displays.

**CRT** This specifies that the compiler listing created by the AUTO procedure displays at the display station that is running the AUTO procedure.

**NOXREF** This specifies that the AUTO procedure does not produce a cross-reference listing of the RPG II program. If you do not specify a parameter, NOXREF is assumed.

**XREF** This specifies that a cross-reference listing is produced.

**mrt maximum**

This specifies the maximum number of display stations (also called multiple requester terminals or MRTs) that can use the program at the same time. Specify any number from 0 through 99. If you do not enter the parameter, a value of 0 is assumed. A value of zero indicates that the program is a single requester terminal (SRT) program. Each display station that runs the program is really running its own copy of the program. A value of 1 or more indicates that the program is a multiple requester terminal (MRT) program.

For more information on MRT programs, see the *System/36 Environment Programming* book.

**NONEP** This specifies that the program is not a never-ending program (NEP). If you do not enter a parameter, NONEP is assumed.

**NEP** This specifies that the program is an NEP. See the NEP parameter of “ATTR OCL Statement” on page 5-8 for more information on NEPs.

**output library**

This specifies the name of the library that is to contain the compiled program. If you do not specify this parameter, the source member library is assumed.

**SOURCE, PSOURCE, or NOSOURCE**

This is supported for System/36 compatibility only. The value is not used.

**DEBUG or NODEBUG**

This is supported for System/36 compatibility only. The value is not used.

**program size**

This is supported for System/36 compatibility only. The value is not used.

**NOHALT** This is supported for System/36 compatibility only. The value is not used.

**HALT** This is supported for System/36 compatibility only. The value is not used.

**REPLACE**

This specifies that if you are creating a program, and a load member or a subroutine member with the same name as your program already exists in the output library, the newly compiled program replaces the existing load or subroutine member. No message appears indicating the replace. If you do not specify a parameter, REPLACE is assumed.

**NOREPLAC**

This specifies that if you create a load member or subroutine member, and a load member or subroutine member with the same name as your program already exists in the output library, a message appears and you can either replace the member or cancel the procedure.

**LINK** This specifies that a program is to be created. If you do not enter a parameter, LINK is assumed.

**NOLINK** This specifies that no program is to be created.

**NOOBJECT**

This is supported for System/36 compatibility only. The value is not used.

**OBJECT** This is supported for System/36 compatibility only. The value is not used.

**subroutine library**

This is supported for System/36 compatibility only. The value is not used.

**GEN** This specifies that if the source program contains a CONSOLE file, the display formats for that CONSOLE file are created. If you do not specify parameter, GEN is assumed.

**NOGEN** This specifies that no display formats for a CONSOLE file are created.

## BASIC

### **work file size**

This is supported for System/36 compatibility only. The value is not used.

### **data dictionary name**

This specifies the library that contains the communications file definition to use with the compiling program. On System/36, the communications file definition was stored in a data dictionary.

### **MRO or NOMRO**

This is supported for System/36 compatibility only. The value is not used.

### **Example**

This example shows how to compile an RPG II program named PAYROLL. The source program is in the current library. The compiled and link-edited load member is placed in the current library. A source listing and a cross-reference listing are generated.

```
AUTOC PAYROLL,,,,XREF,,,,SOURCE
```

---

## BACKUP Procedure

The IBM System/34 BACKUP procedure is not supported. To do similar function, see "SAVELIBR Procedure" on page 4-228 or use the Save Library (SAVLIB) command.

See the *CL Reference* book for more information on the SAVLIB command. See the book, *Backup and Recovery – Advanced*, for more information on how to save and restore your system libraries.

---

## BALPRINT Procedure

Not supported. On System/36, the BALPRINT procedure balanced spooled output among a group of printers by redirecting spool file entries from the busy printers to the idle printers in a printer group.

There is no similar function available on the AS/400 system. You can use multiple output queues, with each output queue directed to a different printer, but the OS/400 operating system will not do balancing of spooled files between the output queues.

See the *Printer Device Programming* book and the *System/36 Environment Programming* book for more information on controlling your spooled output on the AS/400 system.

---

## BASIC Procedure

Not supported. On System/36, the BASIC procedure started a BASIC session.

To do a similar function, use the Start BASIC (STRBAS) command.

See the *CL Reference* book for more information on the STRBAS command. See the *BASIC User's Guide*, SC09-1823 for more information on the AS/400 BASIC licensed program.

---

## BASICP Procedure

Not supported. On System/36, the BASICP procedure ran a BASIC procedure that contained no errors and was saved as a library source member.

To do a similar function, use the Start BASIC Procedure (STRBASPRC) command.

See the *CL Reference* book for more information on the STRBASPRC command. See the *BASIC User's Guide*, SC09-1823 for more information on the AS/400 BASIC licensed program.

---

## BASICR Procedure

Not supported. On System/36, the BASICR procedure ran a BASIC program that contained no errors and was saved as a library subroutine member.

On the AS/400 system, compiled BASIC programs can be run like any other program. You can use either "LOAD OCL Statement" on page 5-55 and "RUN OCL Statement" on page 5-83, or the Call (CALL) command.

See the *CL Reference* book for more information on the CALL command. See the *BASIC User's Guide*, SC09-1823 for more information on the AS/400 BASIC licensed program.

---

## BASICS Procedure

Not supported. On System/36, the BASICS procedure converted a library source member that contained a BASIC program into a library subroutine member.

To do a similar function, use the Create BASIC Program (CRTBASPGM) command.

See the *CL Reference* book for more information on the CRTBASPGM command. See the *BASIC User's Guide*, SC09-1823 for more information on the AS/400 BASIC licensed program.

---

## BASLOAD Procedure

Not supported. On System/36, the BASLOAD procedure installed the BASIC support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## BASSAVE Procedure

Not supported. On System/36, the BASSAVE procedure saved the BASIC support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

### BGUATTR Procedure

BGUATTR procedure

Not supported. On System/36, the BGUATTR procedure changed the printer fill patterns or color palettes used by the Business Graphics Utilities/36 (BGU/36).

To do a similar function, use the Start BGU (STRBGU) command.

See the *CL Reference* book for more information on the STRBGU command. See the *BGU User's Guide and Reference* book for more information on the AS/400 BGU licensed program.

---

### BGUCHART Procedure

Not supported. On System/36, the BGUCHART procedure created or updated a chart, printed a chart on a graphics-capable printer, displayed a chart on a graphics-capable display station, or plotted a chart on a plotter.

To do a similar function, use the Start BGU (STRBGU) command.

See the *CL Reference* book for more information on the STRBGU command. See the *BGU User's Guide and Reference* book for more information on the AS/400 BGU licensed program.

---

### BGUDATA Procedure

Not supported. On System/36, the BGUDATA procedure copied a user-generated graph data input file (GDIF) to a data member and stored the data member in the library specified by the library name parameter.

To do a similar function, use the Start BGU (STRBGU) command.

See the *CL Reference* book for more information on the STRBGU command. See the *BGU User's Guide and Reference* book for more information on the AS/400 BGU licensed program.

---

### BGUGRAPH Procedure

Not supported. On System/36, the BGUGRAPH procedure created or updated a graph, printed a graph on a graphics-capable printer, displayed a graph on a graphics-capable display station, plotted a graph on a plotter, or created a graph object file version of the graph.

To do a similar function, use the Start BGU (STRBGU) command and Display Chart (DSPCHT) commands.

See the *CL Reference* book for more information on the STRBGU and DSPCHT commands. See the *BGU User's Guide and Reference* book for more information on the AS/400 BGU licensed program.

---

## BGULOAD Procedure

Not supported. On System/36, the BGULOAD procedure installed the Business Graphics Utilities/36 (BGU/36) support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## BGUSAVE Procedure

Not supported. On System/36, the BGUSAVE procedure saved the Business Graphics Utilities/36 (BGU/36) support to diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## BLDFILE Procedure

The BLDFILE procedure creates a disk file that contains no data. The file is created in the current files library for resident files or QTEMP for job or scratch files, and can then be referred to as an existing file by following jobs and job steps, which can place data into the file. The BLDFILE procedure can be used to create a file on a remote system when the file name refers to a distributed data management (DDM) file. See the book, *Distributed Data Management*, for more information on creating remote files.

The BLDFILE procedure runs the \$FBLD utility program.

```
BLDFILE file name, { S
                  I
                  D }, { BLOCKS
                       (B)
                       RECORDS
                       (R) }, size, record length, [ A1
                                                    A2
                                                    A3
                                                    A4
                                                    block number ],
[ S
  T ], { key position, key length }, [ NDFILE
                                     DFILE ], [ NODUPKEY
                                               DUPKEY ], [ extend value ]
```

59020657-1

**file name** This specifies the file to be created. A file name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, @, or \$). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. Do not use ALL as a file name.

**S** This specifies that a sequential disk file is to be created.

**I** This specifies that an indexed disk file is to be created.

**D** This specifies that a direct disk file is to be created.

### BLOCKS or B

This specifies that space for the file is to be allocated (or reserved) by blocks.

### RECORDS or R

This specifies that space for the file is to be allocated (or reserved) by records.

**size** This specifies the size of the file to be created. If BLOCKS or B is specified, the size of the file can be from 1 through the maximum number of blocks of disk storage configured on the system. If RECORDS or R is

specified, the size of the file can be 1 through 8 000 000 records. The number cannot be greater than the number of unused blocks or records available for user files. You can use the CATALOG procedure to determine the amount of space available for files.

**record length**

This specifies the length of the record in bytes, and can be any decimal number from 1 through 4096. One byte contains one character of data.

**A1,A2,A3, or A4**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**block number**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**T, J, or S** This specifies the disk file retention type (if the file retention is not specified, T is assumed):

<b>T</b>	Resident file retention type
<b>J</b>	Job file retention type
<b>S</b>	Scratch file retention type

**Note:** For information about file retention types, see "FILE OCL Statement (for Disk Files)" on page 5-25.

**key position**

This specifies the starting position of the key area within the record, and must be a number from 1 through 4096. The key position is required if an indexed file is being created (I is specified for parameter 2). If a key position is specified but an indexed file is not being created, an error message is displayed. The entire key, defined by the key position and key length, must be within the record.

**key length**

This specifies the length of the key area within the record, and must be a number from 1 through 120. The key position is required if an indexed file is being created (I is specified for parameter 2). If a key length is specified but an indexed file is not being created, an error message is displayed. The entire key, defined by the key position and key length, must be within the record.

**DFILE** This specifies that the file is to be a delete-capable file.

**NDFILE** This specifies that the file is not to be a delete-capable file. If this parameter is not specified, NDFILE is assumed.

**DUPKEY** This specifies that duplicate keys are to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored.

**NODUPKEY**

This specifies that duplicate keys are not to be allowed in the indexed file being created. If this parameter is not specified, NODUPKEY is assumed. If the file being created is not an indexed file, this parameter is ignored.

**EXTEND VALUE**

This specifies the value by which a file can be extended each time additional space is needed in the file. The value specified is the number of

records (if the file is allocated in records) or blocks (if the file is allocated in blocks) by which the file is to be extended.

The extend value cannot exceed 8 000 000 records or 312 815 blocks. If a value of 0 is specified, the file being created will not be able to be extended; that is, it will not be automatically extended each time additional space is needed.

If an extend value is not specified, an extend value is calculated. This calculated value is the number of records by which the file is extended each time additional space is needed. This value is calculated by dividing 32 768 by the specified record length. For example, if the specified record length is 100, then  $32\,768/100 = 327$ , and the file will be extended by 327 records.

### Example 1

This example creates a resident, delete-capable file that is 13 blocks long. The file is INVOICE, the record length is 50 bytes, and each record contains a 4-byte key beginning at position 9 in the record. The file then has a calculated extend value of 655 records.

```
BLDFILE INVOICE,I,BLOCKS,13,50,A1,T,9,4,DFILE
```

### Example 2

This example creates a job file named JOB for the PAYROLL procedure. The file is a sequential file, 10 blocks in size, and has a record length of 100. The file then has a calculated extend value of 327 records.

```
* Create a job file
BLDFILE JOB,S,BLOCKS,10,100,,J
* Run the payroll procedure
PAYROLL
```

---

## BLDGRAPH Procedure

The BLDGRAPH procedure is used to build a graphics object file containing the graphic options and parameters for printing with other data print files.

**Note:** On System/36, the BLDGRAPH procedure was part of the Intelligent Printer Data Stream Advanced Functions (PRPQ).

See the *System/36 Environment Programming* book for more information on creating graphics files and merging text and graphics.

```
BLDGRAPH member name,library name,file name
```

59020647-0

### member name

The name of the library source member which contains the graphic options and parameters. Each record in the source member must be in the following format:

Columns 1 through 8, the option name

Column 9, leave blank

Columns 10 through 89, the parameters for the option

**Note:** Comment records are supported by this utility. Place an asterisk (\*) in column 1 of the source records containing the comment.

**library name**

The library name which contains the source member. If not specified, only the current library is searched.

**graphics file name**

The name of the graphics object file to be created.

**BLDINDEX Procedure**

The BLDINDEX procedure creates a resident file that contains the alternative index for a physical file. The physical file must be a resident direct, sequential, or indexed file in the current files library. The alternative index is created in the current files library. The BLDINDEX procedure can be used to create an alternative index file on a remote system when the file name references a distributed data management (DDM) file. See the *Distributed Data Management* book for more information on creating remote files. The physical file that the alternative index is based on must also be a DDM file, and the two DDM files must indicate that the files are at the same remote location. See the *System/36 Environment Programming* book for more information on alternative indexes.

An alternative index file can be thought of as a file that is identical to the original physical file, but the key is defined differently. Note that the physical file is not copied; the keys in the alternative index file provide a different way to access the records in the physical file.

The BLDINDEX procedure runs the \$FBLD utility program.

```

BLDINDEX alternative index file name,key position1,key length1,
      physical file name, [ mddy, [ DUPKEY, [ A1,
                          ddmy, [ NODUPKEY, [ A2,
                          yymm, [ block number,
                          dd ] ] ] ] ]
      [ key position2,key length2 ], [ key position3,key length3 ]
    
```

SP020063-0

**alternative indexed file name**

This specifies the new file that will contain the alternative index for the physical file.

**key position1**

This specifies, for the alternative index, the starting position within the record for the entire key area or the starting position of the first field in the key. The position must be a number from 1 through 4096. The entire key, defined by the key position and key length, must be within the record.

**key length1**

This specifies the total length of the key area or the length of the first field in the key. The key can occupy a single area within the record or

be divided into two or three fields. The total key length of all fields cannot exceed 120 bytes.

**physical file name**

This specifies the physical file for which the alternative index is being created. The physical file must be a resident file.

**mmddy, ddmmy, or yymmdd**

This specifies the date the physical file was created. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If this parameter is not specified and more than one file exists with the name, the latest file is used.

**DUPKEY** This specifies that duplicate index keys are allowed in the alternative index file. If no parameter is entered, DUPKEY is assumed.

**NODUPKEY**

This specifies that duplicate index keys are not allowed in the alternative index file.

**A1, A2, A3, or A4**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**block number**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**key position2**

This specifies, for the alternative index, the starting position of the second field in the key. The position must be a number from 1 through 4096. A key field cannot overlap another key field. If this parameter is not specified, no values may be specified for the third key field or third key length, and the key must occupy a single area within the record.

**key length2**

This specifies, for the alternative index, the length of the second field in the key. The total key length of all fields cannot exceed 120 bytes. A key field cannot overlap another key field. If this parameter is not specified, no values may be specified for the third key field or third key length, and the key must occupy a single area within the record.

**key position3**

This specifies, for the alternative index, the starting position of the third field in the key. The position must be a number from 1 through 4096. A key field cannot overlap another key field. If this parameter is not specified, the key either occupies a single area within the record or is divided into the fields specified by the first two key position and key length parameters.

**key length3**

This specifies, for the alternative index, the length of the third field in the key. The total key length of all fields cannot exceed 120 bytes. A key field cannot overlap another key field. If this parameter is not specified, the key either occupies a single area within the record or is divided into the fields specified by the first two key position and key length parameters.

**Example 1**

This example shows how to create an alternative index file from a physical file named CUSTOMER. The CUSTOMER file is keyed on a customer number in positions 1 through 4 of the record. The alternative index file is to be keyed on the customer's name, which is in positions 5 through 20 of the record (a key length of 16), and is to be named CUSTNAME.

```
BLDINDEX CUSTNAME,5,16,CUSTOMER,,A2
```

**Example 2**

This example shows how to create an alternative index file with a key divided into two fields from the physical file named EMPLOYEE. The EMPLOYEE file is a direct, delete-capable file. The alternative index file is to be keyed on the employee's department number, which is in positions 40 through 44 of the record (a key length of 5), and employee number, which is in positions 1 through 8 of the record (a key length of 8). The alternative index file is to be named EMPLDEPT.

```
BLDINDEX EMPLDEPT,40,5,EMPLOYEE,,,1,8
```

**Example 3**

This example shows how to create an alternative index file with a key divided into three fields from a physical file named INVNTY, which has a creation date of June 25, 1984. The INVNTY file is an indexed file that is keyed on the part number, positions 1 through 8 of the record (a key length of 8). The alternative index file is to be keyed on the date each item of inventory was last ordered, which is in positions 64 through 69 of the record (a key length of 6), the supplier code, which is in positions 70 through 75 of the record (a key length of 6), and the part number. The alternative index file is to be named LOINVTRY.

```
BLDINDEX LOINVTRY,64,6,INVNTY,062584,,A3,70,6,1,8
```

---

**BLDLIBR Procedure**

The BLDLIBR procedure creates a new library and optionally copies a disk, diskette, tape, or tape cartridge file containing one or more library members into the new library. The new library is always created in the system storage pool.

The authority required to access the library is the create authority (CRTAUT) of library QSYS. If not specified on the creation command, objects created into the library will have their authority determined by the system value QCRTAUT.

The file containing members to be copied must have been created by the FROMLIBR procedure or by the \$MAINT utility. See "FROMLIBR Procedure" on page 4-103 or "\$MAINT Utility" on page A-37 for more information about creating this file.

If a sector mode disk file is being copied to a library, the name of the new library must be the same as the name of the library the members were copied from.

The BLDLIBR procedure runs the \$MAINT utility program.

```

BLDLIBR library name, [library size], [directory size], [
A1
A2
A3
A4
block number
],
[
file name
], [
I1
F1
T1
T2
TC
], [
mmddyy
ddmmyy
yymmdd
], [
S1
S2
S3
M1.nn
M2.nn
], [
AUTO
NOAUTO
], [
REWIND
LEAVE
UNLOAD
]

```

59020658-0

**library name**

This specifies the new library. A library name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, @, or \$). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), minus signs (-) and equal signs (=). Do not use #LIBRARY, F1, READER, PRINT, DISK, TAPE, or ALL as a library name.

**library size**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**directory size**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**A1, A2, A3, or A4**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**block number**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**file name** This specifies the file containing one or more library members to be copied into the new library.

**I1** This specifies that the file containing the library members to be copied is on diskette. If a file name is specified but I1, F1, T1, T2, or TC is not, I1 is assumed.

**F1** This specifies that the file containing the library members to be copied is on disk.

**T1** This specifies that the file containing the library members to be copied is on tape in tape drive 1.

**T2** This specifies that the file containing the library members to be copied is on tape in tape drive 2.

**TC** This specifies that the file containing the library members to be copied is on tape in the tape cartridge drive.

**mmddyy, ddmmyy, or yymmdd**

This specifies the date the physical file was created. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If F1 is specified and a date is not specified, the file with the specified name and the most recent creation date is used. If I1 is specified or assumed and a date is

not specified, the program processes the first file in the diskette VTOC with the specified file name.

**S1, S2, or S3**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**M1.nn. or M2.nn**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**AUTO** If specified for tape (unit is T1 or T2) when the tape reel on the original tape drive finishes, the system switches to the other tape drive to continue processing. If the tape reel on the other tape drive is not the same density as the original tape reel, an error message will be issued. If the other tape drive is not available, the system uses the original tape drive.

**NOAUTO** If specified for tape (unit is T1, T2, or TC) when the tape reel on the original tape drive finishes, the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is tape and neither AUTO nor NOAUTO was specified, AUTO is assumed.

If TC is specified for the unit, the AUTO NOAUTO parameter is ignored.

If the unit is diskette (11), the AUTO NOAUTO parameter is supported for compatibility only. If specified, the value will be syntax-checked, but will be ignored when the diskette is processed.

**REWIND** This specifies, for reel-to-reel tape, that the tape rewinds to the load point after processing completes.

For a tape cartridge, REWIND specifies that the tape is positioned to the beginning of the cartridge after processing completes.

**LEAVE** This specifies, for reel-to-reel tape or a tape cartridge, that the tape is left where it was last processed. The next operation to the tape begins at that point.

**UNLOAD** This specifies, for reel-to-reel tape, that the tape rewinds and unloads after processing completes.

For a tape cartridge, UNLOAD specifies that the tape is positioned to the end of the cartridge after processing completes.

REWIND, LEAVE, and UNLOAD are valid only if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge is processed or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, this process is lengthy. You can shorten the preparation time by specifying UNLOAD, because the tape is left at the end of the tape after the last operation completes.

**Example 1**

This example creates a new library called MYLIB.

```
BLDLIBR MYLIB,100
```

**Example 2**

This example shows how to create a new library and copy one or more library members from a diskette file. The library is to be named YOURLIB. The diskette file is named MEMBERS.

```
BLDLIBR YOURLIB,75,15,,MEMBERS
```

---

## BLDMENU Procedure

The BLDMENU procedure generates, from source members, the library members required to display a menu. A menu allows an operator to start a job by selecting a menu option number instead of entering a command or procedure. Up to 24 item numbers (1 through 24) can be defined in a menu.

See the *ADTS/400: Screen Design Aid for the System/36 Environment* book for information about how to build menus. The BLDMENU procedure is equivalent to the Create System/36 Menu (CRTS36MNU) command.

Two types of menus can be created by BLDMENU: free-format and fixed-format. A **free-format menu** is a menu for which the programmer defines the format of lines 3 through 20. A **fixed-format menu** is a menu that is formatted as two 12-item columns.

To display a menu, the operator enters the menu name on the sign on display or uses the MENU command.

The BLDMENU procedure runs the \$BMENU utility program. It also runs the \$MAINT utility program (LIBRLIBR procedure), the \$MGBLD utility program (CREATE procedure), and the \$\$SFGR utility program (FORMAT procedure).

The input to the BLDMENU procedure is:

- A text message source member called a **command text source member**. The command source member is required and contains calls to the commands and procedures to run when the operator selects items from the menu.
- A source member called an **option text source member**. The option text source member is required for free-format menus, but is optional for fixed-format menus. The option text source member defines the text that appears on the menu.

The output from the BLDMENU procedure is:

- A message load member, called a **command text message file**, that BLDMENU creates from the command text source member. BLDMENU places the command text message file in the output library specified on the BLDMENU procedure. This member must be created to display a menu.
- A message load member, called an **option text message file**, that BLDMENU creates from the optional option text source member. This message load member is created only if the option text source member contains System/36 message source statements. BLDMENU places the option text message file in

the output library specified on the BLDMENU procedure. This option text message file is used to create the display format load member for the menu.

- The **display format load member** for the menu. BLDMENU places the display format load member in the output library specified on the BLDMENU procedure. This member must be created to display a menu.
- A listing that contains one of the following:
  - A partial \$SFGR listing containing warning and terminal messages. If no errors are found during the compile of the menu, a \$SFGR listing is not printed. This listing can only be produced if the option text source member contains \$SFGR source statements instead of message source statements.
  - If the option text source member is omitted or contains message source statements:
    - A list of the option numbers and the corresponding statements from the command source member
    - The actual display text that appears on the menu
- A listing from the Create Display File (CRTDSPF) command.

```

BLDMENU menu name, [option text member name], [source member library],
               [current library]

               [load member library], [REPLACE], [KEEP], [FREEFORM], [IGC]
               [current library]

```

59020637-0

### menu name

This specifies the 1- to 6-character name of the menu (the name given to the display format load member for the menu). The names of the command text source member and the command text message file (the name specified in the first line of the command text source member) must be **menu name##**. For example, if the name of a menu is MENU, the name of the command text message file must be MENU##. Because ## is added to the menu name by BLDMENU, you receive an error message if the menu name has more than 6 characters.

### option text member name

This specifies the 1- to 8-character name of the option text source member, if one exists. The name cannot be the same as the menu name if the member contains message source statements. If the name contains message source statements, the name of the option text load member (the name specified in the first record of the option text source member) must be the same as the name of the source member. The option text member name is optional only if you do not request FREEFORM. If you do not specify the text name, BLDMENU uses the information in the command text message file to generate the descriptive text for the items in the fixed-format menu.

**Notes:**

1. If the option text member name is menu name DT (Description Text), the member must contain System/36 message source statements.
2. If you are using the BLDMENU procedure to update a menu created using the screen design aid (SDA), use only the name of the menu for the option text member name parameter. In addition, you must specify the KEEP parameter. For more information, see the *ADTS/400: Screen Design Aid for the System/36 Environment* book..

**source member library**

This specifies the library that contains the source statements. If you do not specify an input library, the current library is assumed.

**Note:** The command and option text source members must be located in a source file called QS36SRC.

**load member library**

This specifies the library that contains the display format load member (the OS/400 display file) for the menu, the command text load member (the OS/400 message file), and, if you specify KEEP, the option text load member (the OS/400 message file). If you do not specify an output library, the current library is assumed.

**Note:** If the input library and the output library are not the same, BLDMENU copies the input source members to the output library when it begins processing. Thus, when BLDMENU runs, the output library must have enough space to contain the two input source members, as well as the two load members created by BLDMENU. BLDMENU removes the source members from the output library before ending.

**REPLACE** specifies that:

- If a display format load member already exists in the output library with the same name as the created menu, BLDMENU automatically deletes the existing member from the output library. The user must be authorized to the load member being replaced.
- You cannot use \$BMENU to replace a nondisplay type of file with the menu display file.
- If a load member that is not a display format load member already exists in the output library with the same name as the created menu, BLDMENU displays an error message. The operator must then decide whether to replace the existing member in the output library or to cancel the job. The existing load member could be a program or a message file.
- If the name of the command text source member or the option text source member is the same as the name of a source member or load member in the output library, BLDMENU automatically replaces the existing members in the output library, whether or not REPLACE is specified.

If you do not specify REPLACE, the following occurs:

- If a load member already exists in the output library with the same name as the created menu, BLDMENU displays an error message. The operator must then decide whether to delete the existing member in the output library or to cancel the job.
- If the name of the command text source member or the option text source member is the same as the name of a source member in the output library, BLDMENU displays an error message and cancels the job.

**Note:** A menu cannot be rebuilt while being displayed at a display station.

**KEEP** This specifies that if you also specify the option text member name parameter, the option text load member created by BLDMENU remains in the output library. If you do not specify KEEP, the option text load member is deleted from the output library before BLDMENU ends.

The \$BMENU utility program only uses the option text load member to create the display format load member and is not needed after the menu is built.

## FREEFORM

This specifies the creation of a free-format menu. If you do not specify FREEFORM, a fixed-format menu is created. If you specify FREEFORM, you must also specify an option text member.

**IGC** This specifies that the system-generated text (for example, the ENTER NUMBER, COMMAND, OR OCL prompt) displays as double-byte characters and you can enter double-byte data into the input field of the menu screen. The IGC parameter is for the double-byte (ideographic) version of the OS/400 program and is only partially supported for nondouble-byte systems. \$BMENU checks the QIGC system value. If the system value is a '0', then the IGC parameter is only partially supported.

### Notes:

1. If you try to print output with double-byte headings on a non-double-byte printer, blanks print in place of the double-byte characters.
2. If the menu screen prints, the READY FOR OPTION NUMBER OR COMMAND prompt may contain extraneous blanks between some of the double-byte characters. However, these blanks are not shown when the menu screen displays.

**Warning:** If you build a menu and use the IGC parameter, a warning message will be displayed if you attempt to display the menu on a nondouble-byte display station.

## Example

The following example creates a command text source member named MENU2## and a menu text source member named MENU2DT. Both members are stored in a library named MYLIB. To build a menu named MENU2 using the BLDMENU procedure, enter the following. The library members created by BLDMENU are placed in a library called MYLIB, and the option text load member created by BLDMENU is deleted from MYLIB before BLDMENU ends.

```
BLDMENU MENU2,MENU2DT,MYLIB,MYLIB,, ,FREEFORM
```

---

## BUILD Procedure

Not supported. On System/36, the BUILD procedure allowed you to display and correct data on the disk after a disk error had occurred.

There is no similar function available on the AS/400 system. You can find out if disk read errors have occurred on your system by using the Work with Problems (WRKPRB) command. You can display or change disk storage by running the *Display/Alter/Dump* service function provided by the Start System Service Tools (STRSST) command.

See the *CL Reference* book for more information on the WRKPRB and STRSST commands. See the *System Operation* book for more information on working with system problems. See the *AS/400 Licensed Internal Code Diagnostic Aids – Volume 1* book for more information on the system service tools (SST).

---

## CACHE Procedure

The CACHE procedure is supported as a no-operation command. Only syntax checking of the procedure parameters is done.

<pre> CACHE [ALTER       START       STOP], [size], [pagesize             2K]           </pre>
--

59020524-1

On System/36, the CACHE procedure allowed the system operator to create, change, or delete the disk **cache**. The **cache** was a buffer used to keep disk data in main storage to reduce the number of accesses to the disk.

On the AS/400 system, this function is not required.

---

## CATALOG Procedure

The CATALOG procedure lists:

- The names and information about database files, remote distributed data management (DDM) files, and save files. If ALL is specified, all the files in the current files library are listed. If a specific file name is specified, the file is located using the library list if the library list search is in effect. If the library list search is not in effect, the file is located in the current files library.
- The names of all libraries on the system.
- The names of all folders and data dictionaries on the system.
- The names and information about diskette files. The CATALOG procedure reads the volume table of contents (VTOC) on the diskette. The diskette VTOC contains an entry for each file on the diskette. The VTOC entry contains information about the file such as the label, creation date, and the location.
- The names and information about tape and tape cartridge files. The CATALOG procedure reads the volume label and header label information on standard labeled tapes and tape cartridges.

You can direct the output from the CATALOG procedure to the system list device or, if the unit parameter is F1, to a disk file. If you specify an output file name, you direct output to that file. Otherwise, output goes to the system list device. To determine your system list device, enter the STATUS SESSION command. See "SYSLIST Procedure" on page 4-253 or "PRINT Procedure" on page 4-178 for information on how to change the system list device.

You must be listed in the system distribution directory to list document folders.

The CATALOG procedure runs the \$LABEL utility program. The CATALOG procedure does not list AS/400 source physical files or physical files with the filetype (\*SRC).

To list *disk* information:

```
CATALOG [ ALL
          file name
          library name
          folder name ] , [ F1 ] , , [ NAME
          LOCATION ] , , [ output file name ]
```

59020068-1

To list *diskette* information:

```
CATALOG [ ALL
          file name ] , I1 , [ starting location
          S1
          S2
          S3
          M1.nn
          M2.nn ] , [ ending location
          starting location
          S1
          S2
          S3
          M1.nn
          M2.nn ] ,
[ NAME
  LOCATION ]
```

59020069-2

To list *tape* or *tape cartridge* information:

```
CATALOG [ ALL
          file name ] , [ T1
          T2
          TC ] , , [ NAME
          LOCATION ] , [ REWIND
          LEAVE
          UNLOAD ]
```

59020070-5

**ALL** This specifies the listing of all file, library, folder, and data dictionary names. ALL is assumed if you do not specify the first parameter.

**file name or library name or folder name or data dictionary name**

This specifies the file, library, folder or data dictionary for which the listing of VTOC information will occur. The label information is listed for tape. If more than one file exists on disk or diskette with the specified name, the listing includes information for all files with the specified name. Only the first tape file with a specified label is displayed.

**F1** This specifies the listing of VTOC entries for the disk. F1 is assumed if you do not specify the second parameter. See "Sample Disk VTOC Listings" on page 4-33 for a description of the items on the listing.

**I1** This specifies the listing of VTOC entries for a diskette. See "Diskette VTOC Listing" on page 4-39 for a description of the items on the listing.

- T1** This specifies the listing of label information for the tape mounted on tape drive 1. See “Tape Label Listings” on page 4-39 for a description of the items on the listing.
- T2** This specifies the listing of label information for the tape mounted on tape drive 2. See “Tape Label Listings” on page 4-39 for a description of the items on the listing.
- TC** This specifies the listing of label information for the tape cartridge. See “Tape Label Listings” on page 4-39 for a description of the items on the listing.

**NAME** This specifies the sorting and listing of entries by name in alphabetic order. NAME is assumed if you do not specify the fifth parameter.

**Note:** This is supported for compatibility only. Disk files are always sorted by name. Diskette and tape files are always listed by location.

#### **LOCATION**

This specifies the sorting and listing of entries by their location sequence on the disk, diskette, or tape.

**Note:** This is supported for compatibility only. Disk files are always sorted by name. Diskette and tape files are always listed by location.

#### **starting location and ending location**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

Valid values for the starting and ending locations are:

- S1, S2, or S3
- M1 or M2
- M1.nn or M2.nn (nn is a magazine slot number from 1 to 10)

**REWIND** This specifies, for reel-to-reel tape, that the tape rewinds to the load point after processing completes.

For a tape cartridge, REWIND specifies that the tape is positioned to the beginning of the cartridge after processing completes.

**LEAVE** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. The REWIND action always takes place.

**UNLOAD** This specifies, for reel-to-reel tape, that the tape rewinds and is unloaded after processing is completed.

For a tape cartridge, UNLOAD specifies that the tape is positioned to the end of the cartridge after processing is completed.

REWIND, LEAVE, and UNLOAD are valid only if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge is processed or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, this process is lengthy. You can shorten the preparation time

by specifying UNLOAD, because the tape is left at the end of the tape after the last operation completes.

## output file name

This specifies the name of the disk file to which to write the output of the CATALOG procedure. The output file has a different format than the System/36 format. The file name can have from one to eight alphameric characters and must not contain blanks, commas, or quotes. ALL is not a valid file name. If you do not specify this parameter, output is written to the SYSLIST device. This creates the specified file as a sequential file on the local system. The size of the listed VTOC determines the size. The record length is 132 bytes. You cannot use this parameter to list diskette or tape information.

**Note:** A file or library with the specified name and the current date must not already exist on the disk. Output to a disk file is only valid when the second parameter is F1.

## Example 1

The following example displays VTOC information for the disk file called PAYROLL:  
CATALOG PAYROLL

## Example 2

The following example lists, by location, the names of the files on a diskette:  
CATALOG ALL,I1

## Example 3

The following example lists, by name, the names of all the files, libraries, folders, and data dictionaries on the disk:  
CATALOG

## Example 4

The following example lists, by location, the names of all files on a tape on tape drive 1 and rewinds the tape when done:  
CATALOG ALL,T1

## Example 5

The following example lists, by location, the names of all the files on a tape on tape drive 2 and always rewinds the tape so that it is positioned at the beginning of the tape.

CATALOG ,T2,,LOCATION,REWIND

## Example 6

The following example lists, by name, the names of all files, libraries, folders, and data dictionaries on the disk, and directs the output to a file named MYLIST:  
CATALOG ,,,,,MYLIST

## Sample Disk VTOC Listings

The following listings show sample disk VTOC printouts. A description of the items on the listing is provided after the example.

### Disk VTOC Listing

```
5763SS1 V3R1M0 940909 Disk VTOC Display 12/14/94 10:08:49 Page 1
IBM OS/400 S/36 Environment CATALOG Listing
Files Library - TEST
Device Capacity - 2968.18 Megabytes / 1159447 Blocks
Space currently available - 295461 blocks
Data Files
Status Codes 1 - Delete capable 2 - Dup keys allowed 3 - Dup keys not allowed 4 - User not authorized
5 - Damaged object 6 - File not available 7 - Dictionary described 8 - On remote system
9 - Externally described
-----File-----
Label Date Org Status Length Used Available Pos Len Records Blocks Parent Member name Last save date
REMOTE 8
DATEDIFF 04/28/86 D 1 32 321 99 420 6 12/08/89
DATEDIFF 04/29/86 D 1 32 321 99 420 6 12/08/89
DATEDIFF 04/30/86 D 1 32 321 99 420 6 12/08/89
DIRECT 05/31/86 D 1 32 321 99 420 6 12/08/89
SEQUENC 03/19/88 S 1024 8 17 25 10 11/29/89
INDEX 01/06/88 I 1,3 226 200 411 3 40 611 54 10/06/89
ALTINDX 01/06/88 X 1,2 226 200 411 NC 10 INDEX 11/15/89
MULTIMEM 11/17/89 PHY 1 120 0 10000 10000 FIRSTM 12/08/89
MULTIMEM 11/17/89 PHY 1 120 5 9995 10000 MBR2 12/12/89
MULTIMEM 11/17/89 PHY 1 120 13 9987 10000 THIRDM 12/09/89
MULTIMEM 11/17/89 PHY 1 120 12 9988 10000 FOURTH 12/08/89
AS400LGL 11/17/89 LGL 1,2,9 21 2 5 AS400PHY MEMBER1 12/06/89
AS400SAV 11/17/89 SAV 528 0
Data files displayed - 14
```

### Disk VTOC Listing by Library

```
5763SS1 V3R1M0 940909 Disk VTOC Display 12/14/94 10:08:49 Page 2
IBM OS/400 S/36 Environment CATALOG Listing
Libraries
-----Library-----
Label Label
#COBLIB #DFULIB
#LIBRARY #RPGLIB
TEST QSSP
QSYS
Libraries displayed - 7
5763SS1 V3R1M0 940909 Disk VTOC Display 12/14/94 10:08:49 Page 3
IBM OS/400 S/36 Environment CATALOG Listing
Document Folders
-----Folder-----
Label Label
QIWSFLRD QIWSMRID
QIWSOS2D
Document folders displayed - 3
5763SS1 V3R1M0 940909 Disk VTOC Display 12/14/94 10:08:49 Page 4
IBM OS/400 S/36 Environment CATALOG Listing
Data Dictionaries
-----Dictionary-----
Label Label
DICTA DICTB
Data dictionaries displayed - 2
* * * * * End of CATALOG LISTING * * * * *
```

## Disk VTOC Listing by Single Name

```

5763SS1 V3R1M0 940909 Disk VTOC Display 12/14/94 09:45:49 Page 1
IBM OS/400 S/36 Environment CATALOG Listing
Files Library - TESTLIB1 Library file found - TESTLIB2
Device Capacity - 2968.18 Megabytes / 1159447 Blocks
Space currently available - 296524 blocks
Data Files
Status Codes 1 - Delete capable 2 - Dup keys allowed 3 - Dup keys not allowed 4 - User not authorized
5 - Damaged object 6 - File not available 7 - Dictionary described 8 - On remote system
9 - Externally described
-----File-----Records----- ---Key--- ----Allocation---- Parent Member Last save
Label Date Org Status Length Used Available Pos Len Records Blocks name date
PR100 02/20/89 S 7 100 100 2 1024 12/08/89
AR100 02/20/89 X 2 100 100 2 1 12 PR100 12/08/89
DICTIONARY -DICT01 DEFINITION - FILEDEF1
***** End of CATALOG listing *****

```

### Description of Items on the Disk VTOC Listing

**Page** This indicates the page number of the printout.

#### Job Name

This specifies the name of the job from which CATALOG runs. This appears only in a listing of job files.

#### Device Capacity

This specifies the size of the disk in megabytes (1 megabyte equals 1 000 000 bytes) and in blocks.

#### Files Library

This is the current default files library as set by the CHGS36 command or the FLIB procedure.

#### Library file found

The System/36 environment can search the library list for an individually cataloged file. If the file is found in a library other than the files library, the name of the library is displayed.

The following fields are shown only for files:

#### File Label

This specifies the label of the file.

**File Date** This specifies the file creation date. A date of 00/00/00 indicates that the file has no members. If the date of 00/00/00 is for an alternative index or logical file, the file is also not associated with any specific physical file member.

**File Org** This specifies how the file is organized:

- S** Sequential
- D** Direct
- I** Indexed
- X** Alternative index

**Note:** If the file is an alternative index file, the allocation size of the file is not listed.

**Blank** A DDM file (when ALL is specified for the first parameter).

**Note:** If the file is a DDM file (remote file) and either ALL is specified for the first parameter or the remote file

does not exist, no information is listed about the file. Use the Display File Description (DSPFD) command to get information about the file.

**PHY** A physical file created using the AS/400 Create Physical File (CRTPF) command with filetype (\*DATA) specified.

**LGL** A logical file created using the AS/400 Create Logical File (CRTLF) command with filetype (\*DATA) specified.

**Note:** If the file is a logical file, the records used, records available, and allocated size of the file are not listed.

**SAV** A save file that may have been created with the System/36 FROMLIBR procedure, the \$MAINT utility, or the AS/400 Create Save File (CRTSAVF) CL command.

**Note:** If the file is a save file, the records available and the allocated size of the file are not listed.

### File Status

This specifies the status of the file, indicating:

- 1 A delete-capable file.
- 2 You can have duplicate keys.
- 3 You cannot have duplicate keys.
- 4 You are not authorized to use this file.
- 5 The file has been damaged and more information is not available.
- 6 Either the file or more information about the file is presently unavailable. This could be because the file is locked (being used) by another program or Free Storage (STG(\*YES)) was specified when this file was saved.
- 7 A file described in a data dictionary.
- 8 A remote file.
- 9 A file described by data description specifications (DDS).

**Note:** File status 7 on the System/36 means that a file format was described in a data dictionary, although the System/36 called the file externally described. The AS/400 system defines the term externally described as a file formatted by DDS. File status 7 in the System/36 environment means that the file is linked to a file format definition in a data dictionary. File status 9 means that the file is described by DDS contained in the file object.

For more information on storage freed and the SAVOBJ command, see the *CL Reference* book. Catalog displays all available information for the file.

### Record Length

This specifies the length of the records in the file. This is a decimal number showing the number of bytes in each record.

## Records Used

This specifies the number of records currently containing data in the file. The number of records used does not reflect any records added by a user currently using the file. Plus signs indicate that the number of records used is greater than 99 999 999.

When a physical file is individually cataloged, the available records of any logical file built over the physical file are not displayed.

## Records Available

This specifies the number of unused records in the file. Plus signs indicate that the number of records used is greater than 99 999 999.

When a physical file is individually cataloged, the available records of any logical file built over the physical file are not displayed.

**Key Pos** This specifies the starting position of the key fields in the record. If the key is divided into two or more fields, NC (not contiguous) is displayed.

For keyed files individually cataloged, the displayed information mentioned above is followed by a list of each individual key position.

Logical files found on the AS/400 system can contain up to 32 different record formats. Each logical file contains individual groups of the file's keys. When logical files are individually cataloged, the key groupings are listed by format.

**Key Len** This specifies the length of the key field in the record. If the key is divided into two or more fields, the total length of the keys in bytes is displayed. Plus signs indicate that the total key length is greater than 999 bytes.

For keyed files individually cataloged, the displayed information mentioned above is followed by a list of each individual key length. Plus signs indicate that the individual key length is greater than 999 bytes.

Logical files found on the AS/400 system can contain up to 32 different record formats. Each logical file contains individual groups of the file's keys. When logical files are individually cataloged, the key groupings are listed by format.

**Note:** Additional lines may be shown for the key position and key length fields if a specific file is listed. For alternative indexed files with keys divided into two or three fields, the starting position and length of the fields that make up the key are shown.

## Allocation Records

This specifies the number of allocated records (in decimal) for a file. The number of records allocated is rounded up to the next higher block. This number is shown only if the file was allocated by records. Plus signs indicate that the number of allocated records is greater than 99 999 999.

## Allocation Blocks

This specifies the number of allocated blocks (in decimal) for a file. Plus signs indicate the file does not have a maximum size. For AS/400 physical and logical files, the number of allocated blocks is not displayed.

**Parent** This specifies the physical file over which the alternative index file was built. The AS/400 physical files can have multiple logical files built over them. If a physical file has multiple logical files built over it, the number of files displays instead of the member names.

To display the logical files built over a physical file, use the Display Database Relationship (DSPDBR) command. To display the physical files built over a logical file, use the Display File Description (DSPFD) command.

For AS/400 physical files that are individually cataloged, the logical files built over them are also displayed. The parent field for these logical files is not displayed.

**Note:** If a physical file is listed with alternatives defined on it, additional lines are displayed that identify its alternative index files.

#### **Member Name**

A file member is a collection of related records in a file. The names of file members are shown only for the AS/400 physical or logical files.

#### **Last save date**

The last save date is the date the data in the member was last saved to media or a save file. The System/36 environment files that have a file organization of D, S, I, or X contain one member. AS/400 files that have a file organization of PHY or LGL can contain multiple members.

**Note:** If a specific file is listed and the file is dictionary-described, an additional line displays the associated data dictionary and definition.

#### **Data Files Displayed**

This indicates the number of files listed.

The following fields are shown for libraries:

#### **Library Label**

This specifies the label of a library.

#### **Libraries Displayed**

This indicates the number of libraries displayed.

The following fields are shown for folders:

#### **Folder Label**

This specifies the label of a folder.

#### **Folders Displayed**

This indicates the number of folders displayed.

The following fields are shown for data dictionaries:

#### **Dictionary Label**

This specifies the label of an IDDU data dictionary.

#### **Data Dictionaries Displayed**

This indicates the number of IDDU data dictionaries listed.

**Note:** If a specific file is listed and that file is externally described, an additional line is displayed, which identifies the associated data dictionary and definition.

Table 4-2 on page 4-38 is a chart indicating the beginning column and the length of the field that you can use for each item on disk VTOC listings.

*Table 4-2. Beginning Columns and Field Lengths for VTOC Listings*

<b>Beginning Column</b>	<b>Field Length</b>	<b>Name of Field</b>
1	7	Product ID
9	2	Version number
11	2	Release number
13	2	Modification number
16	6	Release date
45	8	Session date
54	8	Time
68	4	Page number
26	10	System/36 files library
92	10	Library file found
34	18	Device capacity (megabytes)
76	16	Device capacity (blocks)
47	16	Space available (blocks)
21	8	Job name
50	5	Number of files/libraries/folders/dictionaries displayed
1	10	File label
14	8	File date
24	3	File organization
30	9	File status
39	5	Record length
45	8	Records used
54	8	Records available
65	5	Key position
71	3	Key length
75	8	Allocation records
84	9	Allocation blocks
95	10	Parent
109	10	Member name
121	8	Last save date
19	10	Dictionary
48	10	Definitions
1	12	Library, folder, or dictionary label (1st)
40	12	Library, folder, or dictionary label (2nd)
19	12	Dictionary name
48	12	Definition name

## Diskette VTOC Listing

The listing will be the same as that of the Display Diskette (DSPDKT) command. For more information, see the *CL Reference* book.

## Tape Label Listings

The listings will be the same as that of the Display Tape (DSPTAP) command. For more information, see the *CL Reference* book.

---

## CGU Procedure

The CGU allows you to start a character generator utility (CGU) session.

For more information on the CGU procedure, refer to the *1769* book.

**Note:** The CGU procedure supports extended character tables for Japan, Korea, Taiwan, and the People's Republic of China.

```
CGU [ 24 x 24 printer id ] , [ 18 x 18 printer id ]  
    [ session printer ]    [ session printer ]
```

59020071-0

### 24 x 24 printer id

This is supported for System/36 compatibility only. The value is not used.

### 18 x 18 printer id

This is supported for System/36 compatibility only. The value is not used.

### Example

The following example starts a CGU session:

```
CGU
```

---

## CGULOAD Procedure

Not supported. On System/36, the CGULOAD procedure installed the character generator utility (CGU) support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## CGUSAVE Procedure

Not supported. On System/36, the CGUSAVE procedure saved the character generator utility (CGU) support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

## CHGXLATE Procedure

Not supported. On System/36, the CHGXLATE procedure allowed you to modify the tables that are used to translate characters from ASCII to EBCDIC and from EBCDIC to ASCII.

See the *Client Access/400 for DOS and OS/2 Technical Reference* book for more information on the use of translate tables with Client Access for OS/400.

## CHNGEMEM Procedure

The CHNGEMEM procedure changes the name, subtype, or reference number of a library member. The CHNGEMEM procedure cannot change IBM-supplied library members.

The CHNGEMEM procedure runs the \$MAINT utility program.

```

CHNGEMEM { member name
           member name,ALL
           ALL
           }, { SOURCE
              (S)
              PROC
              (P)
              LOAD
              (O)
              SUBR
              (R)
              LIBRARY
            }, { library name
              current library
            },
           [ newname ], [ subtype ], [ reference number ]

```

59020074-0

### member name

This specifies the changing of the library member.

### member name,ALL

This specifies the changing of the library members whose names begin with the specified characters (member name). Specify up to 7 characters for member name.

**ALL** This specifies the changing of all members of the library. An error message occurs if you specify this parameter with the newname parameter.

### SOURCE or S

This specifies the changing of only the library source members. If you do not specify a parameter, SOURCE is assumed. Source members are all the members in a source file named QS36SRC in the specified library.

### PROC or P

This specifies the changing of only the library procedure members. Procedure members are all the members in a source file named QS36PRC in the specified library.

### LOAD or O

This specifies the changing of only the library load members. Load members are all objects in a library with the following OS/400 object types:

**\*PGM** Programs  
**\*MSGF** Message files

**\*FILE** Display files (attribute of \*DSPF)

**SUBR or R**

This specifies the changing of only the library subroutine members. Subroutine members are all objects in a library with the OS/400 object type of \*PGM (programs).

**LIBRARY** This specifies the changing of all library member types such as SOURCE, PROC, LOAD, and SUBR, including all objects with the following OS/400 object types:

**\*CHTFMT** Chart formats  
**\*CLS** Classes  
**\*CMD** Commands  
**\*DTAARA** Data areas  
**\*FCT** Forms control tables  
**\*FILE** Files (files with all attributes: PF, LF, PRTF, and so on)  
**\*GSS** Graphic symbol sets  
**\*JOB** Job descriptions  
**\*JOBQ** Job queues  
**\*MENU** Menus  
**\*MSGQ** Message queues  
**\*OUTQ** Output queues  
**\*QRYDFN** Query definitions  
**\*SBSD** Subsystem descriptions  
**\*SSND** Session descriptions  
**\*TBL** Tables

**library name**

This specifies the changing of the library containing the members. If you do not specify a library name, the current library is assumed.

**newname**

This specifies the new member name of the changed members. If you specify **member name,ALL**, the new member name must contain the same number of characters as specified in the **member name,ALL** parameter. The remaining characters in each member name do not change.

**subtype** This specifies the subtype of the members to be changed. If not specified, the subtype is not used as a qualifier when selecting members. Valid subtypes are:

Subtype to Enter	Subtype Displayed	Meaning
ARP	rpt or RPT36	The RPG auto report member
ARS	ARS36	The automatic response member
ASM	ASM36	The assembler member
BAP	BASP or BASP36	The BASIC procedure (source member)
BAS	BASP or BASP36	The BASIC member
BGC	BGC36	The business graphics chart
BGD	BGD36	The business graphics data
BGF	BGF36	The business graphics format
COB	CBL or CBL36	The COBOL member
CSM		Communications and System Management member*
CSP		Cross-system product member*
DFU	DFU36	The data file utility member
DLS		Document library service member*
DTA	DTA36	The data member
FMT	DSPF or DSPF36	The display format member
FOR	FOR36	The FORTRAN member
ICF		CONFIGIC procedure member*
KEY		KEYS procedure member*
MNU	MNU or MNU36	The menu member
MSG	MSGF or MSGF36	The message member
PHL	PHL36	The phone list member
QDE		Query data entry member*
QRY		Query member*
RPG	RPG or RPG36	The RPG member
SRT	SRT36	The sort member
SSP		CNFIGSSP procedure member*
TXT	TXT	Text member
UNS	UNS36	Unspecified
WSU	WSU36	The work station utility member
X25		X.25 packet switching control link*

\* These subtypes are supported for compatibility only. No OS/400 subtype corresponds to these System/36 subtypes. If one of these subtypes were specified, no members would be found.

#### reference number

This specifies the user-defined reference number to which the members are to change. The value specified must be numeric and within the range of 0 to 999 999. The *subtype* and *reference number* for load (LOAD) and subroutine (SUBR) members cannot be changed. If a specific member name is specified, an error message is issued. If ALL is specified for the first parameter and the member type is LOAD or SUBR, an error is issued. If ALL and LIBRARY are specified, the load and subroutine members will not be changed and no error message is issued. For more information, see "Reference Numbers for Library Members" on page 1-18.

**Example 1**

The following example changes the names of all the members that begin with PAY to DAY. The members are in a library called PAYROLL.

```
CHNGEMEM PAY,ALL,LIBRARY,PAYROLL,DAY
```

**Example 2**

The following example changes the subtype and reference number of a source member named PAY01 in a library named PAYROLL. The new subtype is SRT with a new reference number of 123.

```
CHNGEMEM PAY01,SOURCE,PAYROLL,,SRT,123
```

**CLRPF Procedure**

The CLRPF procedure is supported only for compatibility with the IBM System/36. Only the procedure interface is supported. See the Clear Physical File Member (CLRPFM) command in the *CL Reference* book for information on how to clear the contents of a physical file member. The CLRPF procedure does not change the creation date of the file member that is cleared.

**CNFIGICF Procedure**

Not supported. On System/36, the CNFIGICF procedure configured or set up a communications subsystem that consisted of a line member and a subsystem member. Communications subsystems were required for the Interactive Communications Features (SSP-ICF), MSRJE, 3270 device emulation, C & SM, and document distribution.

A similar function is available from the Configure Devices and Communications (CFGDEVCMN) menu.

See the *System/36 Environment Programming* book for more information on the AS/400 communications configuration. See the *System/36 to AS/400 Migration Aid User's Guide and Reference* book for more information on migrating System/36 configurations to the AS/400 system.

**CNFIGSSP Procedure**

Not supported. On System/36, the CNFIGSSP procedure configured or set up the System/36 programming support, system environment, display station attributes, and display station configuration.

On the AS/400 system, a similar function is available by using the following menu or command:

- To configure local and remote display stations and printers, use the Configure Devices and Communications (CFGDEVCMN) menu.
- To manage IBM AS/400 licensed programs, use the Manage License Programs (LICPGM) menu.
- To define system values, use the Change System Value (CHGSYSVAL) command.

## COBOLC

See the *Local Device Configuration* book for more information on configuring local and remote devices on the AS/400 system. See the *System/36 to AS/400 Migration Aid User's Guide and Reference* book for more information on migrating System/36 configurations to the AS/400 system. See the *Software Installation* book for more information on installing licensed programs. See the *CL Reference* book for more information on the CHGSYSVAL command. See the *Work Management* book for a complete description of system values.

---

## CNFIGX25 Procedure

Not supported. On System/36, the CNFIGX25 procedure allowed you to define the network configuration, logical channel configuration, and virtual circuit configuration.

A similar function is available from the Configure Devices and Communications (CFGDEVCMN) menu.

See the *System/36 Environment Programming* book for more information on AS/400 communications configuration. See the *System/36 to AS/400 Migration Aid User's Guide and Reference* book for more information on migrating System/36 configurations to the AS/400 system.

---

## COBLOAD Procedure

Not supported. On System/36, the COBLOAD procedure installed the COBOL support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## COBOL Procedure

This procedure is supported only for compatibility with the IBM System/34. See "COBOLC Procedure" for information about compiling COBOL programs.

---

## COBOLC Procedure

The COBOLC procedure compiles a COBOL/36 source program. For more information on COBOL, see the book, *System/36-Compatible COBOL User's Guide and Reference*.

```
COBOLC  source member name, [ source member library ], [ output member library ],
      [ current library ], [ source member library ],

      [ NODSM ], [ PRINT ], [ XREF ], [ NONEP ], [ mrt maximum ], [ work file size ],
      [ DSM ], [ NOPRINT ], [ NOXREF ], [ NEP ], [ 0 ], [ 40 ],
      [ SOURCE ], [ DEBUG ], [ MAP ], [ OFFSET ], [ OBJECT ],
      [ NOSOURCE ], [ NODEBUG ], [ NOMAP ], [ NOOFFSET ], [ NOOBJECT ],

      [ copy from library name ], [ subroutine member library ],

      [ data dictionary name ], [ LINK ], [ NOMRO ],
      [ NOLINK ], [ MRO ]
```

59020659-0

**source member name**

This specifies the source member to compile.

**source member library**

This specifies the name of the library that contains the source member. If you do not specify this parameter, the current library is assumed.

**output member library**

This specifies the name of the library to contain the load member or subroutine member produced by the compile. If the library is not specified on the PROCESS statement or if this parameter is used, the source member library is assumed.

**NODSM** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**DSM** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**PRINT** This specifies that the compiler output prints. If you do not specify a parameter, PRINT is assumed.

**NOPRINT** This specifies that no compiler output is displayed or printed.

**CRT** This specifies that the compiler output is displayed.

**XREF or NOXREF**

This specifies whether a cross-reference listing of the COBOL program is produced. If you do not specify a parameter, the option specified in the COBOL program's PROCESS statement is assumed.

**XREF** This specifies that a cross-reference listing is produced.

**NOXREF** This specifies that no cross-reference listing is produced.

**NONEP** This specifies that the program is not an NEP. If you do not specify a parameter, NONEP is assumed.

**NEP** This specifies that the program is an NEP. For more information about NEPs, see the *System/36 Environment Programming* book.

**mrt maximum**

This identifies the program being compiled as a multiple requester terminal (MRT) program and specifies the maximum number of requester display stations that can be using the program at one time. The maximum value is 99. If the value is 0 or is not specified, the program is not a MRT program.

**work file size**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**SOURCE or NOSOURCE**

This specifies the print option to use instead of the print option entered in the COBOL program's PROCESS statement. If you do not specify a parameter, the option specified in the PROCESS statement is used.

**SOURCE** This specifies that the compiler produces a listing of the source statements in the COBOL program along with any error messages.

**NOSOURCE**

This specifies that the compiler does not produce a listing of the source statements in the COBOL program. Only the error messages are listed.

**DEBUG or NODEBUG**

This specifies the debug option to use instead of the debug option entered in the COBOL program's PROCESS statement. If you do not specify a parameter, the option specified in the PROCESS statement is used.

**DEBUG** This specifies that the compiler uses the COBOL debug facility.

**NODEBUG**

This specifies that the compiler does not use the COBOL debug facility.

**MAP or NOMAP**

This specifies whether the COBOL program's Data Division mapping option is used instead of the option entered in the COBOL program's PROCESS statement. If you do not specify a parameter, the option specified in the PROCESS statement is used.

**MAP** This specifies that the compiler maps the program's Data Division and object program.

**NOMAP** This specifies that the compiler does not map the program's Data Division and object program.

**OFFSET or NOOFFSET**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**OBJECT or NOOBJECT**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**copy from library name**

This specifies the library to search when a COBOL COPY statement is encountered. If you do not specify this parameter, the library specified in the COBOL program's PROCESS statement is assumed. See the *System/36-Compatible COBOL User's Guide and Reference* book for more information.

**subroutine member library**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**data dictionary name**

This specifies the library name that contains the communications file definition to use with the program being compiled.

**LINK** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**NOLINK** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**MRO or NOMRO**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**Example**

The following example compiles a COBOL program. The source member is named PAYROLL and it is in a library named TESTLIB. The compiled program is named PAYROLL and is placed in a library named PAYLIB. The DSM option is syntax-checked and no further action is taken.

```
COBOLC PAYROLL,TESTLIB,PAYLIB,DSM,,XREF,,,,SOURCE,DEBUG
```

---

**COBOLCG Procedure**

The IBM System/34 COBOLCG procedure is not supported.

See “COBOLC Procedure” on page 4-44 for information about compiling COBOL programs. See “LOAD OCL Statement” on page 5-55 and “RUN OCL Statement” on page 5-83 for information about running programs.

---

**COBOLG Procedure**

The IBM System/34 COBOLG procedure is not supported.

See “LOAD OCL Statement” on page 5-55 and “RUN OCL Statement” on page 5-83 for information about running programs.

---

**COBOLONL Procedure**

Not supported. On System/36, the COBOLONL procedure allowed you to develop a COBOL program. You could create a new program, or make a large number of changes to an existing program. The COBOLONL procedure showed a series of displays that allow you to enter, compile, and change COBOL programs.

Similar functions are provided by the S/36 Programming Languages Procedures (S36PGMLNG) menu. This menu provides options for running the COBOLC, COBSDA, and COBSEU procedures. See “COBOLC Procedure” on page 4-44, “COBSDA Procedure” on page 4-48, and “COBSEU Procedure” on page 4-48 for more information.

---

**COBOLP Procedure**

Not supported. On System/36, the COBOLP procedure caused a menu to be displayed that allowed you to select the option you wanted to perform. You could enter, compile, or change COBOL programs.

Similar functions are provided by the S/36 Programming Languages Procedures (S36PGMLNG) menu. This menu provides options for running the COBOLC, COBSDA, and COBSEU procedures. See “COBOLC Procedure” on page 4-44, “COBSDA Procedure” on page 4-48, and “COBSEU Procedure” on page 4-48 for more information.

## COBSAVE Procedure

Not supported. On System/36, the COBSAVE procedure saved the COBOL support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

## COBSDA Procedure

The COBSDA procedure starts the screen design aid (SDA) procedure. See the *ADTS/400: Screen Design Aid* book for information about display formats and about how to use SDA.

See the *System/36-Compatible COBOL Reference Summary* book for more information about this procedure and about COBOL.

```
COBSDA
```

59020083-0

The COBSDA procedure has no parameters.

### Example

The following example starts the COBSDA procedure:

```
COBSDA
```

## COBSEU Procedure

The COBSEU procedure starts the source entry utility (SEU) procedure. This allows you to create or change a COBOL program or procedure. See the *System/36-Compatible COBOL User's Guide and Reference* book for more information about this procedure and about COBOL. See the *ADTS/400: Source Entry Utility* book for more information on SEU.

```
COBSEU member name, [ S ], [ seu format member ], [ statement length ],
                    [ library name
                      current library ]
```

59020084-0

### member name

This specifies the source member or procedure that you want to create or change.

**S** This specifies a COBOL source member. If you do not specify a parameter, S is assumed.

**P** This specifies a library procedure member.

### seu format member

This is supported for System/36 compatibility only. The value is not used.

**statement length**

This is supported for System/36 compatibility only. The value is not used.

**library name**

This specifies the library that contains or will contain the library member being changed or created. If you do not specify a library name, the current library is assumed.

**Example**

The following example starts COBSEU to create or change a procedure named PAYROLL. The procedure is contained in the current library.

```
COBSEU PAYROLL,P
```

**COMPRESS Procedure**

The COMPRESS procedure is supported as a no-operation command. Only syntax checking of the procedure parameters is done.

On System/36, the COMPRESS procedure collected all unused disk space within the user area. This allowed you to make room available on the disk for larger files, libraries, and folders by gathering the smaller unused areas together.

On the AS/400 system, this function is not required.

See the *System/36 Environment Programming* book for more information on managing your files and libraries. See the book, *Planning for and Setting Up OfficeVision/400* for more information on folder management on the AS/400 system.

```
COMPRESS [ ALL ] , [ FREELOW ]
          [ A1 ]   [ LOW ]
          [ A2 ]   [ FREEHIGH ]
          [ A3 ]   [ HIGH ]
          [ A4 ]
```

59020089-0

**CONDENSE Procedure**

The CONDENSE procedure is supported as a no-operation procedure for libraries and is not supported for folders.

On System/36, the CONDENSE procedure collected all unused space within a library or folder into a single area. This allowed you to make room available for more members or documents by gathering the smaller unused areas together.

If you are condensing a library, the system checks the syntax of the procedure parameters and verifies that a library by the specified name does exist.

If you are condensing a folder, use the Reorganize Document Library Object (RGZDLO) command.

See the *CL Reference* book for more information on the RGZDLO command, and the *System/36 Environment Programming* book for more information on managing

## COPYDATA

your libraries. Also, see the *Planning for and Setting Up OfficeVision/400* book for more information on folder management on the AS/400 system.

```
CONDENSE [ library name  
          current library  
          folder name ] , [ FOLDER ]
```

59020090-0

---

## COPYDATA Procedure

The COPYDATA procedure copies the contents of a data file on disk to another data file on disk. During the copy operation, the COPYDATA procedure can:

- Create the new output file in either the same file organization or a different file organization. Files can be sequential, direct, or indexed.
- Omit or include specific records.
- Remove deleted records.
- Change the record length.
- Reorganize data records in an indexed file.
- Limit the number of output records to be copied.

The COPYDATA procedure can also be used with physical, database data files which do not have a System/36 environment direct, indexed or sequential file organization. When such files are used, the new file may be created with file attributes that differ from those you requested or expected. A message allowing you to continue is sent whenever this may occur. See the *System/36 Environment Programming* book for more information on using System/36 environment files and AS/400 files.

You can use the COPYDATA procedure to copy files to or from a remote system. The AS/400 system allows either the old file or the new file to be on a remote system, but not both. The COPYDATA procedure cannot create a copy of an externally described file on a remote system.

The COPYDATA procedure can process a file being used by another job on the system, providing that job's FILE OCL statement specifies DISP-SHRRR or DISP-SHRRM. See "FILE OCL Statement (for Disk Files)" on page 5-25 for more information about file sharing.

The COPYDATA procedure cannot copy alternative index files.

The COPYDATA procedure runs the \$COPY utility program.

```

COPYDATA old file name, [ mddyymmddyy ], new file name, { BLOCKS, blocks
                        [ ddmmyy ] [ yymmdd ] { RECORDS, records }

[ A1 ] [ T ] [ NORREORG ] [ INCLUDE ] [ position ] [ EQ ]
[ A2 ] [ J ] [ REORG ] [ OMIT ] [ ] [ NE ]
[ A3 ] [ ] [ ] [ ] [ ] [ ] [ LT ]
[ A4 ] [ ] [ ] [ ] [ ] [ ] [ LE ]
[ block number ] [ ] [ ] [ ] [ ] [ ] [ GE ]

[ 'characters' ] [ record length ] [ maximum records ] [ SAME ]
[ ] [ ] [ ] [ ] [ ] [ ] [ S ]
[ ] [ ] [ ] [ ] [ ] [ ] [ I ]
[ ] [ ] [ ] [ ] [ ] [ ] [ D ]

[ key position ] [ key length ] [ DUPKEY ]
[ ] [ ] [ ] [ ] [ ] [ ] [ NODUPKEY ]

```

59020091-1

**old file name**

This specifies the file to copy. The file can be direct, indexed or sequential. Alternative index files cannot be copied. See "BLDINDEX Procedure" on page 4-20 for more information about alternative indexes.

**mmddyymmddyy, ddmmyy, or yymmdd**

This specifies the creation date of the file to copy. The date, if specified, must be in the session date format. Use the STATUS SESSION command to determine the session date format. If you do not specify a date and more than one file exists with the same file name, the most recent file created is copied.

**new file name**

This specifies the name of the new file containing copied records from the old file. The first character of the file name must be alphabetic (A through Z, #, @, or \$). The file name can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. Do not exceed eight characters.

If your system supports date-differentiated files, any files that you create that have the same label as existing files must also have the same attributes as the existing files.

**BLOCKS or RECORDS**

This specifies the size for the new file. If you specify neither BLOCKS nor RECORDS, the size of the new file is the same as the size of the old file. If the file's organization and/or record length will be different than that of the old file, the new file is made large enough to hold the total number of records in the old file.

**BLOCKS** This specifies the number of disk file blocks to allocate (or reserve) for the new file. One disk block contains 2560 bytes.

**Blocks** specifies the number of blocks needed for the new file. It cannot be greater than the configured maximum.

Use the CATALOG procedure to determine the number of disk blocks available for a file. A file cannot be created if it contains more than 16 777 200 records.

**RECORDS**

This specifies the number of records to allocate (or reserve) for the new file. If you specify RECORDS, the system allocates the required

number of blocks to contain those records. The total space reserved for the new file is rounded up to the next block, allowing the file to contain at least the number of records specified. The smallest disk file unit that you can reserve is one block.

**Records** specifies the number of records to allocate for the new file. It cannot be greater than the value 8 000 000.

**A1, A2, A3, or A4**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**block number**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**T**

This specifies that the new file is to be a resident file. A resident file is usually used more than once. The disk area containing a resident file becomes available for another file after the DELETE procedure is used to remove the file from the disk. For more information about resident files, see "FILE OCL Statement (for Disk Files)" on page 5-25.

If no parameter is specified, T is assumed.

**J**

This specifies that the new file is to be a job file. A job file can only be used by following job steps in a job. The job file is defined only within the job and is deleted by the system when the job ends.

Once a job file is created, following steps in the job can refer to the file by using a FILE OCL statement that contains either the RETAIN-S or RETAIN-J parameter, or by not specifying a RETAIN parameter. If you do not specify the RETAIN parameter, or if you specify RETAIN-J, the file still exists as a job file for following job steps. If you specify RETAIN-S for a job file, the job file becomes a scratch file, and is removed from the disk when that job step ends. For more information about job files, see "FILE OCL Statement (for Disk Files)" on page 5-25.

**NOREORG**

This specifies that all deleted records are copied in the copy process. Also, the records in an indexed file are organized in the same position in the new file as they were in the copied file. Deleted records, however, are not copied when you specify any additional parameters for sequential or indexed files. If you do not specify a parameter, NOREORG is assumed.

**REORG**

This specifies that any deleted records in the file to be copied are not copied to the new file. However, deleted records are always copied for direct files. If the file to be copied is an indexed file, the records in the old file are read sequentially by key.

**INCLUDE or OMIT**

This specifies whether specific records in the file to be copied are to be included in or omitted from the new file. The INCLUDE and OMIT parameters work with the position, EQ, NE, LT, GT, LE, GE, and 'characters' parameters. If only 'characters' or position are specified, INCLUDE and EQ are assumed.

You cannot use INCLUDE or OMIT, position, EQ,NE, LT, LE, GT, GE, or 'characters' during a direct-to-direct file copy operation because the

use of these parameters can result in the loss of record positioning in the direct output file.

- position** This specifies, for each record, the first character to compare with the comparison characters. The position can be any number from 1 through 4096. If you do not specify a position and INCLUDE and EQ have been specified (or can be assumed), then every position in the record is compared with the comparison characters until the condition is met.
- EQ** This specifies that the record is to be included in or omitted from the new file if the characters in the record indicated by position are the same as the comparison characters.
- NE** This specifies that the record is to be included in or omitted from the new file if the characters in the record indicated by position are not the same as the comparison characters.
- LT** This specifies that the record is to be included in or omitted from the new file if the characters in the record indicated by position are less than the comparison characters.
- LE** This specifies that the record is to be included in or omitted from the new file if the characters in the record indicated by position are less than or the same as the comparison characters.
- GT** This specifies that the record is to be included in or omitted from the new file if the characters in the record indicated by position are greater than the comparison characters.
- GE** This specifies that the record is to be included in or omitted from the new file if the characters in the record indicated by position are greater than or the same as the comparison characters.

**'characters'**

This specifies the comparison characters. Up to 30 characters can be specified, and the characters should be enclosed by apostrophes ('). You can include blanks and commas (,) but you cannot specify apostrophes as data.

**record length**

This specifies the record length of the new file, and is any number from 1 through 4096. If you do not enter this parameter, the record length of the file to be copied is used for the record length of the new file.

If the record length of the old file is less than the specified record length of the new file, the additional record positions in the new file are filled with blanks. If the record length of the old file is greater than the specified record length of the new file, the extra positions are truncated. If the new file is an indexed file and the key field would be truncated, an error message is displayed.

**maximum records**

This specifies the total number of records to copy into the new file. Enter any number greater than 0.

**Note:** The system cannot create a file that contains more than 16 777 200 records.

- SAME** This specifies that the new file is to have the same organization as the file to be copied. If you do not specify a parameter, SAME is assumed.
- S** This specifies that the new file is to be organized as a sequential file.
- I** This specifies that the new file is to be organized as an indexed file.
- D** This specifies that the new file is to be organized as a direct file.

**key position**

This specifies the starting position of the key for the new file. You must specify the key position if the new file is an indexed file, but the old file was not organized as an indexed file. The key position can be any number from 1 through 4096. If you do not specify a value, the key position for the indexed file being copied is assumed. The entire key, defined by the key position and key length, must be within the record. If you enter either the key position or key length, you must also enter the other.

**key length**

This specifies the length of the key for the new file. You must specify the key length if the new file is an indexed file, but the old file was not organized as an indexed file. The key length can be any number from 1 through 120. If you do not specify a value, the key length for the indexed file being copied is assumed. The entire key, defined by the key position and key length, must be within the record. If you enter either the key position or key length, you must also enter the other.

**DUPKEY** This specifies that duplicate keys are allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If you do not specify this parameter, the attribute of the input file is the output file attribute.

**NODUPKEY**

This specifies that duplicate keys are not allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If you do not specify this parameter, the attribute of the input file is the attribute of the output file.

**Note:** NODUPKEY does not remove duplicate key records.

**Example 1**

The following example copies FILE1 and creates an exact copy of it, FILE2:

```
COPYDATA FILE1,,FILE2
```

**Example 2**

The following example copies a file named FILE1, which is an indexed file, and creates a new file named FILE3: The new file is an indexed file and the key is in positions 5 through 24.

```
COPYDATA FILE1,,FILE3,,,,,,,,,,,,I,5,20
```

**Example 3**

The following example shows how to remove deleted records from a delete-capable sequential file named FILEA:

1. FILEA is copied into a temporary file named TEMP and is reorganized using the COPYDATA procedure.
2. FILEA is removed from the disk using the DELETE procedure.
3. The temporary file is then renamed as FILEA using the RENAME procedure.

The following procedures are used:

```
COPYDATA FILEA,,TEMP,,,,REORG
DELETE FILEA,F1
RENAME TEMP,FILEA
```

**Example 4**

The following example copies only the records that contain the characters 'CHICAGO, IL' starting in position 45 from FILE1 into a new file, FILE4:

```
COPYDATA FILE1,,FILE4,,,,,INCLUDE,45,EQ,'CHICAGO, IL'
```

**Example 5**

The following example copies, from FILE1 to FILE5, only those records that have the words 'SAMPLE DATA' in them. Note that because no position is specified along with INCLUDE and EQ, a record is copied no matter where the phrase appears.

```
COPYDATA FILE1,,FILE5,,,,,INCLUDE,,EQ,'SAMPLE DATA'
```

**Example 6**

The following example copies, from FILE1 to FILE6, only those records that have a value greater than '0002' in positions 1 through 4:

```
COPYDATA FILE1,,FILE6,,,,,INCLUDE,1,GT,'0002'
```

**COPYDIAG Procedure**

Not supported. On System/36, the COPYDIAG procedure copied a single diagnostic (microcode) diskette.

To do a similar function, use the Duplicate Diskette (DUPDKT) command.

See the *CL Reference* book for more information on the DUPDKT command.

**COPYI1 Procedure**

The COPYI1 procedure copies the files from one diskette onto other diskettes. Use COPYI1 to create a copy of a diskette. When using the COPYI1 procedure, consider the following:

- The output diskette must be in the same format as the diskette being copied, either a diskette 1 or a diskette 2D, and initialized in the same format (either FORMAT or FORMAT2, which are parameters for the INIT procedure).

- Diskettes with important permanent files are the diskettes normally copied. Because diskettes can wear out as they are used, it is a good idea to have more than one copy of important diskette files.
- You cannot display and correct unreadable sectors.
- You cannot copy a single file from one diskette to another.
- The system does not check for the specified volume ID on the diskette before it makes the copy. The volume ID and owner ID from the input diskette are written to the output disk.
- When you specify the DELETE parameter, files on the diskette that have been deleted are not copied, but expired files are copied.

The COPY11 procedure runs the \$DUPRD utility program.

```
COPY11 [ ALL
        File name ], [ mmddy
                  ddmmy
                  yymdd ], volume id, [ DELETE ], [ PRESERVE ], [ copies ],
[ input slot location ], [ output slot location ], [ CHECK ]
```

59020092-0

**ALL** This specifies the copying of all files on a diskette to another diskette. If you do not specify the first parameter, ALL is assumed.

**file name** This is not supported. If a single file name is specified, an error message is issued.

**mmddy, ddmmy, or yymdd**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**volume id**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. The output volume ID and OWNERID are overwritten to the ID and OWNERID of the input diskette.

**DELETE** This specifies that deleted files (files whose VTOC entries are erased) are not copied. Expired files are copied.

**PRESERVE**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**copies** This specifies the number of diskette copies to make, and can be any number from 1 through 99. If you do not specify a copies value, 1 copy is assumed.

**input slot location**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

Valid values for the input slot location are:

- S1, S2, or S3
- M1 or M2
- M1.nn or M2.nn (nn is a magazine slot number from 1 to 10)

**output slot location**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

Valid values for the output slot location are:

- S1, S2, or S3
- M1 or M2
- M1.nn or M2.nn (nn is a magazine slot number from 1 to 10)

**CHECK** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**Example**

The following example copies all the files on the input diskette onto another diskette:

```
COPYI1 ALL
```

---

## COPYPRT Procedure

The COPYPRT procedure creates a new disk file and copies one or more spool files to it. You can copy up to 255 spool files at one time.

Use the COPYPRT procedure to display or print the contents of an existing disk file created by a previous COPYPRT procedure.

The COPYPRT procedure runs the \$UASF and \$UASC utility programs.

<pre> COPYPRT [ ALL          spool id          Fxxxx          SYSTEM          NOCOPY         ] , [ file name ] , [ RELEASE                           CANCEL                         ] , [ CRT                           PRINT                         ] </pre>
--

59020093-1

**ALL** This specifies the copying of all of a user's spool files. If you do not specify the first parameter, ALL is assumed. The only spool files copied are those which are not being processed by spool and those spool files having the same user ID as the operator that entered the procedure.

**spool id** This specifies the 6-character spool file ID of the spool file to copy. If the spool file is being processed by spool, it is not copied.

**Fxxxx** This specifies the forms number of the one or more spool files to copy. **xxxx** is the 4-character forms number of the spool files to copy. If the spool file is being processed by spool, it is not copied.

For more information about forms numbers, see "PRINT Procedure" on page 4-178, "FORMS OCL Statement" on page 5-46, or "PRINTER OCL Statement" on page 5-68.

**SYSTEM** This specifies the copying of all spool files not being processed by spool.

**NOCOPY** This specifies the printing of the copied spool files in the user file. If you specify NOCOPY, then RELEASE or CANCEL are invalid.

**file name** This specifies the name of the user file into which the spool files are to be copied, or the user file from which the copied spool files are to be displayed or printed. You must specify a file name if you specify ALL, SYSTEM, or NOCOPY as the first parameter.

If you do not specify a file name but you do specify a spool ID or forms number, the file name assumed is either the spool ID or Fxxx and the 2-character ID of the display station. For example, if the operator at display station W3 enters the COPYPRT procedure specifying a spool ID of SP0032, the file name is SP0032W3. If the operator at display station W3 has a forms number of 0001, the file name is F0001W3.

The format of the file is described in "File Format for COPYPRT Files" on page 4-59.

## RELEASE

This specifies that, after the copy is complete, the spool file is released for printing.

**CANCEL** This specifies that, after the copy is complete, the spool is removed from the system.

**CRT** This specifies the Work with Copied Spool Files display that appears at your work station. You can also use this option to print the copied entries. Figure 4-1 on page 4-62 and Figure 4-2 on page 4-62 show how the entries appear and explain the fields on the display. CRT is assumed if the first parameter is NOCOPY and the fourth parameter is omitted.

**PRINT** This specifies the printing of the spool file entries. You can specify PRINT if you also specify NOCOPY.

If you specify this parameter, each copied spool file in the user file is printed. The values for FORMSNO, COPIES, ALIGN, LINES, LPI, CPI, FONT, and JUSTIFY that were used to create the original spool entry are preserved.

If run from a work station, the printer allocated is the session printer. Otherwise, the system printer is used.

Printer requests can be evoked or placed on the job queue.

## Example 1

The following example copies spool file entry SP0001 into a file named SP0001W2. The COPYPRT procedure is entered from display station W2.

```
COPYPRT SP0001
```

## Example 2

The following example copies all spool file entries not being processed by the spool into a file named SPFILE:

```
COPYPRT SYSTEM,SPFILE
```

**Example 3**

The following example copies all spool file entries with forms number 0017 into a file named F0017W3. The COPYPRT procedure is entered from display station W3.

```
COPYPRT F0017
```

**Example 4**

The following example copies ALL spool file entries for a specific user into a file named TEMP. After each spool file entry is copied, it is removed from the system. After all spool file entries have been copied, the information about the copied spool file entries are shown at the user's display station.

```
COPYPRT ,TEMP,CANCEL,CRT
```

**Example 5**

The following example prints all of the copied spool file entries in an existing file named TEMP:

```
COPYPRT NOCOPY,TEMP,,PRINT
```

**File Format for COPYPRT Files**

The record length of the disk file created by the COPYPRT procedure is determined from the largest possible page width in the spool files to be copied. The page width is calculated as you copy the spool files. It is not the page width shown when the spool file attributes are displayed; it is the page width used when the spool file is printed.

Use the following table to determine the record length of the disk file:

Calculated Page Width	Disk File Record Length
1 through 132	150
133 through 198	215
Greater than 198	248

The calculated page width can be determined by:

1. Calculating the size of the print line in inches when the spool file was created. Divide the page width used when the spool file was created by the characters per inch (cpi) used when the spool file was created.
2. Calculating the page width when the file is to be printed. Multiply the print line in inches by the cpi to be used when the spool file is printed.

Normally the cpi used when the spool file is created and the cpi used when the spool file is printed are the same, so the page width used when the spool file is created and the page width calculated for the printing are the same. However, the following conditions can change the cpi used when printing, thus changing the page width used when printing or copying a spool file:

- If the application that generates the spool file uses the advanced printer functions (APF), the cpi can be changed after the printer file is opened.
- The printer file used to create the spool file has the computer output reduction (COR) specified for page rotate (PAGRRT(\*COR)).

- A PRINTER OCL statement with ROTATE-COR is specified for the printer file that created the spool file.
- A PRINTER OCL statement with CONTINUE-YES is specified for the printer file that created the spool file and one of the application programs specified a cpi that was different from the cpi used when the printer file was opened.

The page width used when a spool file is created might differ from the calculated page width when an RPG application is run to create a printed report. The printer file used for the report is P2. Printer file P2 is defined to have a cpi of 10 and the page width is 132. COR was specified for the page rotation when the printer file was created. Because COR was specified for page rotate, the cpi used for printing is 13.3, while the cpi used when the spool file was created is 10. The calculated page width for printing would be 176  $((132/10) \times 13.3 = 176)$ .

If the spool file created by this example was copied to a disk file using the COPYPRT procedure, the record length of the disk file would be 215, not 150.

One way to determine the calculate page width is to display the spool file and check the scale at the top of the display. It is set based on the calculated page width.

One header record is created for each entry in the spool file copied to the disk file. The header record contains the fields shown in the following table:

Beginning Column	Field Length	Contents or Description
1	1	The letter H (to indicate the header)
4	6	The spool ID of the entry
12	8	The procedure name
22	8	The job name
32	8	The user ID of the operator (first 8 characters)
42	8	The printer file name (first 8 characters)
52	2	The printer ID
56	4	The forms number (first 8 characters)
61	2	The number of copies (in binary)
65	2	The number of pages (in binary)
69	4	The number of records (in binary)
74	2	The number of lines per page (in binary)
78	1	The letter I if this entry contains print records with IGC data (otherwise a blank)
81	1	The letter M if this entry contains print records with a length greater than 132 (otherwise a blank)
84	1	Lines per inch (in binary)
85	1	Characters per inch (in binary)
86	1	Font ID (in binary)
87	1	Justify value (in binary)
88	1	Align (Y or N)

<b>Beginning Column</b>	<b>Field Length</b>	<b>Contents or Description</b>
89	1	Constant of zero (in binary)
90	1	Record length (in binary)
92	10	User ID of operator
102	10	Printer file name
112	10	Forms number

All positions not specified above are set to blanks.

The print records follow the header record. These records have a record length of 150, 215, or 248. The print records contain the fields shown in the following table:

<b>Beginning Column</b>	<b>Field Length</b>	<b>Contents or Description</b>
1	2	The page number (in binary)
3	2	The line number (in binary)
5	4	The record number (in binary)
9	1	The letter I if this print record contains IGC data
10	1	An IGC shift-out character (hex 0E) if this print record starts with IGC data
11	132, 198, or 225	The data to print

All positions not specified above are set to blanks.

You can press the Help key to get information about the Work with Copied Spool Files displays. These displays show a list of spool file headers for spool files that were copied to a user file in an earlier job.

```

Work with Copied Spool Files

User file . . . . : EFNREF

Type options, press Enter.
5=Display 6=Print

-----Printer-----
Opt  SP-ID  Proc   User      Name      ID   Line  Formtype
SP3453 CATALOG SOMEONE  $SYSLIST  P1   132  0001
SP3454             SOMEONE  QPTAPDSP  P1   132  *STD
SP3457             SOMEONE  QSYSPRT   P1   132  *STD
SP3458 HELP     SOMEONE  $SYSLIST  P1   132  0001
SP3459 HELP     SOMEONE  $SYSLIST  P1   132  0001
SP3460 LISTLIBR SOMEONE  $SYSLIST  P1   132  0001

More...

Command
===>
F3=Exit      F4=Prompt   F5=Refresh  F9=Retrieve
F11=Display detail information  F12=Cancel

```

Figure 4-1. Work with Copied Spool Files Display

All the information for the copied spool files cannot be presented on one display. Therefore, the information is continued on a second display. You can switch between the two displays using the F11 key.

```

Work with Copied Spool Files

User file . . . . : EFNREF

Type options, press Enter.
5=Display 6=Print

-----Total-----
Opt  SP-ID  Pages  Records  IGC  Lines  Lines  Characters
      SP-ID  Pages  Records  IGC  Per   Per   Per Inch  Font  Justify
SP3453 1      12     N      66   6     10    0      0
SP3454 1       6     N      51   6     10    0      0
SP3457 5      115    N      51   6     10    0      0
SP3458 1       1     N      66   6     10    0      0
SP3459 2       77    N      66   6     10    0      0
SP3460 2       77    N      66   6     10    0      0

More...

Command
===>
F3=Exit      F4=Prompt   F5=Refresh  F9=Retrieve
F11=Display general information  F12=Cancel

```

Figure 4-2. Using F11 on the Work with Copied Spool Files Display

## Description of Fields

**User file** This specifies the name of the user that contains the copied spool files presented on the display.

**Opt** This specifies the option number of the operation you wish to perform.  
The valid options are:

**Option 5** This displays certain copied spool files at the work station.

**Option 6** This prints certain copied spool files. The spool files are copied, not printed, to an output queue associated with the session printer.

The values used for FORMSNO, ALIGN, LINES, LPI, CPI, FONT, and JUSTIFY are those values defined for the current session printer, *not* the values that were used when the original spool file was created

A secondary prompt display appears in which you can supply required print information about the spool files selected for print.

**SP-ID** This specifies the spool ID for the copied spool file.

**Proc** This specifies the name of the procedure executed to create the copied spool file. If this field is blank, the spool file was created outside the System/36 environment.

**User** This specifies the user ID of the person that executed the job that created the spool file.

### Printer Name

This specifies the name of the printer on which the copied spool file was to print.

### Printer ID

This specifies the ID of the printer on which the copied spool file was to print. If this field is blank, the spool file was created outside the System/36.

### Printer Line

This specifies the length of the largest print line in the copied spool file.

### Formtype

This specifies the forms number assigned to this copied spool file.

### Total Pages

This specifies the number of pages for the copied spool file.

### Total Records

This specifies the number of print records in the copied spool file. This number does not include any transparent mode print records that may have been in the spool entry prior to copying. If the number is greater than 9 999 999, '+++++++' displays.

**IGC** This specifies whether the copied spool file contains any double-byte character set (DBCS) characters (also called ideographic [IGC] characters). The following are the valid values:

**Y** This file contains DBCS characters.

**N** This file does not contain DBCS characters.

**Lines Per Page**

This specifies the lines per page of the copied spool file.

**Lines Per Inch**

This specifies the lines per inch of the copied spool file.

**Characters Per Inch**

This specifies the characters per inch of the copied spool file.

**Font**

This specifies the font identifier of the copied spool file. The value displayed is a decimal number.

**Justify**

This specifies the percentage of the copied spool file output to right-justify. The valid values are:

- 0**            The output is not to be right-justified.
- 50**           The output is to be right-justified to a position halfway between the end of the text and the right margin.
- 100**          This output is to be right-justified to the right margin.

**DISPLAY COPIED SPOOL FILE**

The display that appears when you select option 5 on the Work with Copied Spool Files display. A copied spool file is identical to the display presented by the Display Spool File (DSPSPLF) command. See the *CL Reference* for more information on the DSPSPLF command.

**PRINT COPIED SPOOL FILES**

If you select option 6 (Print) on the Work With Copied Spool Files display, one of the following displays appears:

You can press the Help key to get information about the following Print Copied Spool File displays:

```
Print Copied Spool File
Spool ID . . . . . : SP9631
Type information for printing, press Enter.
From page . . . . . 1      1-99999
Through page . . . . . 2      1-99999
Number of copies . . . . . 1      1-255

F3=Exit  F12=Cancel
```

```

          Print Copied Spool File with Warning
Spool ID . . . . : SP9724
Session printer lines per page . . . . . : 66
Copied spool file lines per page . . . . . : 48
Type information for printing, press Enter.
Print spool file . . . . . Y      Y=Yes
                                   N=No
From page . . . . . 1      1-99999
Through page . . . . . 2      1-99999
Number of copies . . . . . 1      1-255

F3=Exit  F12=Cancel

```

You can change the printing defaults on the Print Copied Spool File display. The display appears for each spool file selected for printing (option 6 on the Work with Copied Spool File display).

The display has the following two versions:

- **Without a warning.** This display appears when the lines per page value specified for the session printer is the same as the lines per page of the copied spool file.
- **With a warning.** This display appears when the lines per page value specified for the session printer is different that the lines per page of the copied spool file. If you proceed with printing after the warning issues, the system defaults to the lines per page specified for the session printer.

## Description of Fields

**Spool ID** This specifies the spool ID of the copied spool file that is selected for printing. This value displays for reference only.

### Session printer lines per page

This specifies the lines per page for the session printer. This is the printer on which the copied spool file prints. The lines per page value displays as a comparison to the lines per page of the copied spool file.

### Copied spool file lines per page

This specifies the lines per page of the copied spool file. The lines per page value displays as a comparison to the lines per page of the session printer.

### Print spool file

This specifies whether to continue printing the copied spool file. If printing continues, the session printer's lines per page value is used, and incorrect printing can result. The valid values are:

- Y** Y (yes) specifies printing of the copied spool file.
- N** N (no) specifies no printing of the copied spool file.

# CREATE

## From page

This specifies the page number at which printing begins. You can specify any decimal number from 1 through the number of pages in the spool file. If you specify a value, the value must be equal to or less than the **through page** value. If you do not specify a value, printing begins at the first page of the copied spool file.

## Through page

This specifies the page number after which printing ends. You can specify any decimal number from 1 through the number of pages in the spool file. If you specify a value, the value must be equal to or greater than the **from page** value. If you do not specify a value, one copy of the copied spool file prints.

## Number of copies

This specifies the number of copies to print. Enter a number from 1 through 255. If you specify no entry, 1 is assumed.

**Note:** Default values are set when the display appears.

---

## CREATE Procedure

The CREATE procedure generates a message load member (OS/400 message file) from a message source member. The CREATE procedure is equivalent to the Create System/36 Message File (CRTS36MSGF) command. A message member contains messages that a program can display or print, display on formats, or display with the “ERR Procedure” on page 4-93.

You can also use “// \* (Informational Message) Statement” on page 3-44 and “// \*\* (System Console Message) Statement” on page 3-45 to display messages. You can use “?Mmic? or ?M'mic,position,length'? (Message Member) Expression” on page 3-20 to get a message from a message load member.

The input to the CREATE procedure is a message source member. The message source member contains three types of statements:

- The message control statement
- One or more message text statements
- One or more comment statements (optional)

These statements are described in “Message Member Statements” on page 4-67. If you have the double-byte character set version of the OS/400 program, refer to “Considerations for the Double-Byte Character Set Version of the OS/400 Program” on page 4-70.

You can create the source member using the source entry utility (SEU). SEU is described in detail in the *ADTS/400: Source Entry Utility* book. You can also use the \$MAINT utility program to create source members. See “Create Source or Procedure Members” on page A-39 for more information.

The CREATE procedure runs the \$MGBLD utility program.

```
CREATE source member name, [REPLACE], [library name  
current library], [HALT  
NOHALT]
```

59020094-0

**source member name**

This specifies the existing message source member that contains the message source statements.

**REPLACE**

This specifies that the created message load member replaces an existing load member with the same name. The name of the load member is specified by the message control statement in the source member.

If you do not specify REPLACE, a load member with the same name is not replaced. If a load member with the same name exists, a message displays and the operator can either replace the load member or cancel the job.

REPLACE replaces any type of load member (including an OS/400 message file, program, or display file). You must be authorized to the load member being replaced.

**library name**

This specifies the library that contains the source member and into which the load member is placed. If you do not specify a library name, the current library is assumed.

**HALT** This specifies that if errors are encountered in the message source member, a message displays. If you do not specify a parameter, HALT is assumed.

**NOHALT** This specifies that no message displays if errors are encountered in the message source member, but the return code is set to 2034.

**Example**

The following example generates a message load member from a source member named MESSAGES that is contained in library MYLIB. Enter the CREATE procedure as follows:

```
CREATE MESSAGES,,MYLIB
```

## Message Member Statements

The following describes message member statements.

**The Message Control Statement**

The message control statement specifies the name of the message load member created by the CREATE procedure, and whether the message member is a first- or second-level message member. The message control statement must be the first line in the source member, and only one is allowed.

The format of the message control statement is:

<pre>load member name, <math>\left[ \begin{array}{l} 1 \\ 2 \end{array} \right]</math></pre>
--

59020095-0

**load member name**

This specifies the name of the message load member to create. A member name is up to 8 characters long and must begin with an alpha-

betic character (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). Avoid using the following characters because these have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), and equal signs (=). Do not use DIR, LIBRARY, or ALL as a member name.

- 1 This specifies that the member is a first-level message member. First-level message members can have up to 75 characters of message text. If you do not specify the message level parameter, 1 is assumed.
- 2 This specifies that the member is a second-level message member. Second-level message members can have up to 225 characters of message text.

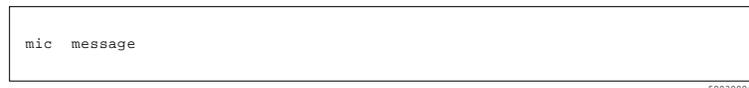
The following example shows the format of a message control statement. The name assigned to the load member is MESSAGES. The load member is a first-level message member.

```
MESSAGES,1
```

### Message Text Statement

The message text statement specifies a message identification code (MIC) and the message text for a message.

The format of the message text statement is:



**mic** This specifies the message identification code (MIC) of the message. The MIC must be a 4-digit number from 0000 through 9999. Place the MIC in positions 1 through 4 of the message text statement. The MICs in a message member must be in ascending order. Specify the same MIC more than once for statements that have a large amount of text. The number of statements that can specify the same MIC is limited to the number of statements required to specify up to 75 characters for first-level messages and up to 225 characters for second-level messages.

**message** This specifies the text of the message. The text area begins at position 6 in the text statement and continues through the end of the source member line. The text for the message is the characters starting in position 6 to the last nonblank character in the message.

This example shows the format of a message text statement. The MIC number is 0014. The text of the message follows the number. For example:

```
0014 This is a message
```

If the message is continued on another source line, the text area on the following line is added to the end of the previous line, including one or more blanks from the previous line. For example:

```

          Column Indicators
          1         2         3         4
1234567890123456789012345678901234567890

```

```

MESSAGE,2
0001 This is the first message and itbbb
0001 is continued on this line.

```

Statements are read up to their record length. In a source member having a record length of 40, the preceding lines produce a second-level message load member named MESSAGE. Message 0001 contains the following message (note the spaces in the combined message caused by the spaces following that first line of message text):

```

This is the first message and it   is continued on this line.

```

**Note:** The RCDLEN source attribute specifies the record length used for records in the source member. If the RCDLEN attribute is not set, the default is the record length of the QS36SRC file minus 12. If the specified RCDLEN is less than the default RCDLEN, characters beyond the RCDLEN position are ignored. The RCDLEN source attribute can be set or changed with the Change System/36 Source Attributes (CHGS36SRCA) and Edit System/36 Source Attributes (EDTS36SRCA) commands. Also, the Restore System/36 Library Members (RSTS36LIBM) command sets the RCDLEN attribute when it restores the source member.

The message text can contain replacement text fields. These fields allow the creation of a single message and the specification of replacement data within the message. For example, you can specify a file name, a user ID, or the time of day when the message is issued. Multiple fields can exist and can be placed anywhere within the message. Use pound symbol (#) characters as place holders in the message text to indicate where the replacement text fields are located. Specify the text used to replace the # characters as the text parameter on “ERR Procedure” on page 4-93. See the *System/36 Environment Programming* book for further information and examples of using replacement text in displayed messages.

### Comment (\*) Statement

The comment statement is optional. The format of the comment statement is as follows:

```

* comment

```

59020097-0

The asterisk (\*) must be in the first position in the statement. This statement does not become part of the message load member. The comment statement cannot be the first statement in the message source member.

### Example Message Source Member

Assume a message source member contains the following statements. Note how the messages for the two programs are grouped.

## CREATE

```
MESSAGES,1
* The following messages are for the program PAYROLL (00xx)
0001 Enter yesterday's date:
0002 Enter today's date:
0003 Enter tomorrow's date:
* The following messages are for the program ACCTS (01xx)
0101 Accounts Payable program is running.
0102 Make sure printer P3 is powered on.
0103 Accounts Payable program is complete.
```

## Considerations for the Double-Byte Character Set Version of the OS/400 Program

For systems with the double-byte character set version of the OS/400 program, you can create a message member that has either double-byte character set or non-double-byte character set messages, or a message member that contains both double-byte character set and non-double-byte character set messages.

If the message member contains only non-double-byte character set characters, you create the source member the same way as for a system without the double-byte character set version of the OS/400 program.

If the message member contains only double-byte character set messages, the source member has a MIC of A000 preceding the message text statements. For example:

```
MESSAGE,2
* double-byte character set Message Member
A000
0001 (double-byte character set text)
0002 (double-byte character set text)
.
.
.
9999 (double-byte character set text)
```

If the message member contains both double-byte character set and non-double-byte character set messages, the source member has a MIC of A000 following the non-double-byte character set messages and preceding double-byte character set messages. The MICs for each type of message can be the same. For example:

```
MESSAGE,2
* non-double-byte character set Message Portion
0001 (non-double-byte character set text)
0002 (non-double-byte character set text)
.
.
.
9999 (non-double-byte character set text)
* double-byte character set Message Portion
A000
0001 (double-byte character set text)
0002 (double-byte character set text)
.
.
.
9999 (double-byte character set text)
```

If an operator is signed on to a double-byte character set session, messages are retrieved from the double-byte character set portion of the message member. If the MIC is not found in the double-byte character set portion, the message is retrieved from the non-double-byte character set portion of the message member.

If a job is evoked or placed on the input job queue, or the operator is not signed on to a double-byte character set session (see "IGC Procedure" on page 4-119), the messages are retrieved from the non-double-byte character set portion of the message member.

For the double-byte character set version of the OS/400 program, special conditions exist when you continue message text from one message text statement to another. If a shift-in character is in one of the last two positions in a line and if a shift-out character is in the sixth position of the continued line, the shift-in/shift-out pair is removed along with any blanks between the pair when the lines are joined in the CREATE procedure.

---

## CSALL Procedure

Not supported. On the System/36, the CSALL procedure allows a user to end the workstation session for a specific workstation or for all workstations, except the system console, in order to perform dedicated system activities. Such dedicated system activities do not exist on the AS/400 system.

---

## DATE Procedure

The DATE procedure changes the session date or the program date.

If you enter the DATE procedure and it is not between a LOAD OCL statement and a RUN OCL statement, the session date changes. A session begins when an operator signs on and ends when the operator enters the OFF command. If you do not use the DATE procedure or DATE OCL statement to establish a session date, the system date in the system value QDATE is the session date.

If you enter the DATE procedure between a LOAD OCL statement and a RUN OCL statement in a procedure, DATE specifies the program date (the date the program uses). The program date is also called the job step date. When the program ends, the session date is the program date. If you do not enter the DATE procedure between a LOAD statement and a RUN statement, the session date is the program date. If you enter two or more DATE procedures between a LOAD statement and a RUN statement, the last DATE procedure is used.

The DATE procedure processes a DATE OCL statement.

### Notes:

1. The program date determines the file retention period for diskette files used by the program. It can be printed on the printed output. The program date is also the creation date of disk, diskette, and tape files created by the program.

The session and program dates are also affected by the CHGJOB CL command. For more information on handling dates in a System/36 environment job see the section that discusses the job date and date format in the *System/36 Environment Programming* book.

## DATE

2. If a job is placed on the input job queue, the program date when the job was placed on the queue is assigned to the job.
3. If 2400 hours (midnight) occurs, the system date automatically updates, but the session date and the program date do not.
4. The DATE procedure affects the AS/400 job date. The job date is changed to the new session date when the DATE procedure is outside a LOAD/RUN pair. When the DATE procedure is between the LOAD and RUN OCL statements, the job date is changed to the new program date. After the RUN OCL statement is processed, the job date is changed back to the session date.



The date specified on the DATE procedure must be in the current session date format. The session date can be in any of three formats: month-day-year (mmdyy), day-month-year (ddmyy), or year-month-day (yyymdd). You can use the STATUS SESSION command to show the current session date format, and the SET procedure to change the current session date format.

**mm**, **dd**, and **yy** must each be a 2-digit number, but you can omit leading zeros in month and day when you use punctuation. You cannot use a date of all zeros (000000). You can enter the date with or without punctuation. For example, you can specify July 24, 1984, in any of the following ways:

```
7-24-84  mm-dd-yy
24-7-84  dd-mm-yy
84-7-24  yy-mm-dd
072484   mmdyy
240784   ddmyy
840724   yyymdd
```

In the punctuated form, you can use any characters except commas (,), quotes ('), numbers, and blanks as punctuation. However, if the DATE procedure is within a procedure, use question marks (?), slashes (/), equal signs (=), plus signs (+), and hyphens (-) with caution because these characters have special meanings within procedures.

### Example 1

The following examples entered by the operator from the keyboard specify a session date for July 1, 1984:

```
DATE 7-1-84
```

or:

```
DATE 070184
```

### Example 2

The following example specifies a job date for the PAYROLL and PAYPRNT programs. The DATE procedure is placed before the first LOAD OCL statement in the procedure.

```

// * 'ENTER PAYROLL DATE'
DATE ?1R?
// LOAD PAYROLL
// FILE NAME-EMPLOYES
// FILE NAME-CHECKS,RETAIN-J,BLOCKS-100
// RUN
// LOAD PAYRNT
// FILE NAME-CHECKS,RETAIN-J
// PRINTER NAME-PRINT,SPOOL-NO,LINES-25,FORMSNO-CHCK,
//          ALIGN-YES,DEVICE-P3
// RUN

```

---

## DEFINEID Procedure

Not supported. On System/36, the DEFINEID procedure was used to specify a list of the remote IDs for a switched communications line using the SSP-ICF BSCEL subsystem. When the System/36 using BSCEL was connected to a remote system by a switched communications line, this list was examined for a comparison with the ID received from the remote system. This list was only examined when multiple remote switched line IDs were specified when the subsystem was configured. If the ID was found, line initialization was successfully completed. If it was not found, the remote system was disconnected.

A similar function is available from the Configure Devices and Communications (CFGDEVCMN) menu. You can use the Work with Controller Descriptions (WRKCTLD) command specifying \*CMN for the controller description (CTLD). From the Work with Controller Descriptions display you can create or change a controller description specifying the remote identifiers (RMTID parameter) for switched BSC communications. You can also use the Create Controller Description (BSC) (CRTCTLBSC) and the Change Controller Description (BSC) (CHGCTLBSC) commands to define remote identifiers (RMTID parameter) for switched BSC communications.

See the *CL Reference* book for more information on the CRTCTLBSC, CHGCTLBSC, and WRKCTLD commands. See the *System/36 Environment Programming* book for more information on AS/400 communications configuration.

---

## DEFINEPN Procedure

Not supported. On System/36, the DEFINEPN procedure provided a way to create or update a phone list for the autocal feature.

On the AS/400 system, one phone number for each controller description may be specified for switched lines.

To do a similar function, use the Work with Controller Descriptions (WRKCTLD) command specifying \*CMN for the controller description (CTLD) parameter. From the Work with Controller Descriptions display you can create or change a controller description specifying a phone number for the connection number (CNNBR) parameter.

See the *CL Reference* book for more information on the WRKCTLD command. See the *System/36 Environment Programming* book for more information on the AS/400 communications configuration.

---

## DEFINLOC Procedure

Not supported. On System/36, the DEFINLOC procedure allowed you to set up a list of names and location IDs of remote locations that you allowed to call your subsystem. (This procedure was valid only for an asynchronous subsystem using X.25 support.)

A similar function is available from the Configure Devices and Communications (CFGDEVCMN) menu. You can use the Work with Configuration Lists (WRKCFGL) command, specifying \*ASYNCLC for the controller type (TYPE) parameter. From the Work with Configuration Lists display you can create, change, or delete a list of asynchronous remote location names and remote location identifiers. You can also create a configuration list of asynchronous remote location names and remote location identifiers using the Create Configuration List (CRTCFGL) command, specifying \*ASYNCLC for the controller type (TYPE) parameter. You can change and delete a configuration list using the Change Configuration List (CHGCFGL) and the Delete Configuration List (DLTCFGL) commands.

See the *CL Reference* book for more information on the CRTCFGL, CHGCFGL, DLTCFGL, and WRKCFGL commands. See the *System/36 Environment Programming* book for more information on the AS/400 communications configuration.

---

## DEFINX21 Procedure

Not supported. On System/36, the DEFINX21 procedure provided a way to create or update a list of public data network connection numbers for the X.21 feature or a short hold mode line configuration.

On the AS/400 system, one phone number may be specified for each controller description. Use the Work with Controller Descriptions (WRKCTLD) command, specifying \*CMN for the controller description (CTLD) parameter. From the Work with Controller Descriptions display you can create or change a controller description specifying a phone number for the connection number (CNNNBR) parameter. Use the Work with Line Descriptions (WRKLIND) command to create or change a line description for the controller. You can specify the minimum time to wait before attempting to dial a number again, using the short delay (SHORTDLY) parameter.

On the AS/400 system, X.21 short-hold mode parameters may be specified for each SDLC line description. Use the Work with Line Descriptions (WRKLIND) command to create or change an SDLC line description. The following X.21 short-hold mode parameters allow other controls:

### **CALLNBR**

This specifies the local telephone number.

### **SHORTRTY**

This specifies the total number of times an attempt should be made to call a particular telephone number.

### **SHORTDLY**

This specifies the minimum time to wait before attempting to dial a number again.

**SHMALLTMR**

This specifies the short-hold mode inactivity timer.

See the *CL Reference* book for more information on the WRKCTLD command.  
See the *System/36 Environment Programming* book for more information on the AS/400 communications configuration.

**DEFINX25 Procedure**

Not supported. On System/36, the DEFINX25 procedure allowed you to create and edit a list of remote network addresses used for the X.25 feature. The DEFINX25 procedure also allowed you to create or edit a list of phone numbers used with the rotary dial function when using the asynchronous subsystem.

On the AS/400 system, one phone number may be specified for each controller description.

To do a similar function, use the Work with Controller Descriptions (WRKCTLD) command, specifying \*CMN for the controller description (CTLD) parameter. From the Work with Controller Descriptions display you can create or change a controller description specifying a phone number for the connection number (CNNBR) parameter.

You can also use the Create Configuration List (CRTCFGL) command, specifying Asynchronous Network Address List (\*ASYNCADR) for the controller type (TYPE) parameter, to create a list of remote network addresses to be used with the rotary dial function. You can change, remove, print, or display a configuration list using the Work with Configuration Lists (WRKCFGL) command.

See the *CL Reference* book for more information on the WRKCTLD, CRTCFGL, CHGCFGL, and DLTCFGL commands. See the *Asynchronous Communications Programming* book for more information on the rotary dial function. See the *System/36 Environment Programming* book for more information on the AS/400 communications configuration.

**DEFSUBD Procedure**

The DEFSUBD procedure creates or deletes a subdirectory or presents a subdirectory display. Always use the folder name whether using either the CREATE or DELETE option, or no option. Use the subdirectory name with only the CREATE and DELETE options.

DEFSUBD	[ CREATE DELETE ]	,folder name,	[subdirectory name]
---------	----------------------------	---------------	---------------------

59020596-0

**CREATE** This specifies the creation of a subdirectory.

**DELETE** This specifies the deletion of a subdirectory.

**folder name**

This specifies the name of the folder.

# DELETE

## subdirectory name

This specifies the name of a subdirectory.

## Example

The following example indicates how to show the Work with Subdirectory display for folder MYFLDR:

```
DEFSUBD ,MYFLDR
```

---

## DELETE Procedure

The DELETE procedure removes one or more files, libraries, or folders from a diskette or disk. The DELETE procedure will delete a file on a remote system if the file name references a distributed data management (DDM) file.

You cannot use the DELETE procedure to do the following:

- Delete a file, library, or folder while it is being used by another job or if you specified a file statement within the job step when the file name and file label for that statement are different.
- Delete a file if you have specified a file statement with JOB=YES for the file within the job.
- Delete a file that has one or more alternative index files on disk.
- Delete the following libraries: #LIBRARY, QSYS, QSSP, QTEMP, or your System/36 environment files library.
- Delete data dictionaries.
- Delete job or scratch files.

To remove one or more members from a library, see "REMOVE Procedure" on page 4-192.

The DELETE procedure runs the \$DELET utility program. For information on how to delete *all* files, libraries, and folders from disk, refer to "\$DELET Utility" on page A-29.

You must be enrolled in the system distribution directory to delete folders.

To delete one or all files from *diskette*:

```
DELETE {file name}, [I1], [SCRATCH REMOVE ERASE], [mddy ddmm yyymmdd], , [starting location],  
S1  
S2  
S3  
M1.nn  
M2.nn  
  
[ending location  
starting location  
S1  
S2  
S3  
M1.nn  
M2.nn], [volume id]
```

S9020103-0

To delete a file, library, folder, or a group of files, libraries, or folders from *disk*:

DELETE	{ file name library name folder name ALL }	,F1,	{ SCRATCH REMOVE ERASE }	{ mddy ddmmy yyymm }	{ LIBR FOLDER ALL }	, , , ,	{ group name }
--------	---	------	--------------------------------	----------------------------	---------------------------	---------	----------------

59020104-0

**file name, library name, or folder name**

This specifies the file to delete from diskette or the file, library, or folder to delete from disk.

**ALL** This specifies the following:

- If you specify I1, all the files on diskette are deleted.
- If you specify F1, you must specify a group name. All the members of the file group are deleted.
- If you specify both F1 and LIBR, and you also specify a group name in the ninth parameter position, all files and libraries that are members of the file group are deleted.
- If you specify both F1 and FOLDER, and you also specify a group name in the ninth parameter position, all files and folders that are members of the file group are deleted.
- If you specify both F1 and ALL, and you also specify a group name in the ninth parameter position, all files, libraries, and folders that are members of the file group are deleted.

**I1** This specifies the file is deleted from diskette. If you do not specify a second parameter, I1 is assumed. If the file is contained on more than one diskette, a message may be displayed and the operator may have to insert the next diskette. If you specify I1 and you specify ALL as the file name, a message is displayed saying that the end of the diskette has been reached. You can delete all files from another diskette with the same volume ID by inserting it in the slot and taking option 0.

**F1** This specifies the file, library, or folder to delete from the disk.

**SCRATCH**

This specifies that if the file is on *diskette*, the expiration date is set to the current job step date. The file remains on diskette and you can access the file only for reading. If you want to reuse free space on a diskette, this space must follow the last active file. If you write any new information onto the diskette, the file no longer exists.

SCRATCH is ignored on a DELETE from disk. Data is always deleted when you delete the file, library, or folder.

If you do not specify a third parameter, SCRATCH is assumed.

**REMOVE** This specifies, for diskette, the removal of the VTOC entry for the diskette file. This removal allows the space in the diskette VTOC to be used for other data. If you want to reuse free space on a diskette, this space must follow the last active file.

REMOVE is ignored on a DELETE from disk. Data is always deleted when you delete the file, library, or folder.

**ERASE** This specifies the removal of the VTOC entry for the diskette file. This removal allows use of the space in the diskette VTOC for other data. If you want to reuse free space on a diskette, this space must follow the last active file. Also, the data contained in the deleted file is removed. That is, zeros replace all characters in the file.

## DELETE

### **mmddy, ddmmy, or yymmdd**

This specifies the creation date of the file to delete. For a disk file, specify the date in the same format as the session date. For a file on diskette, specify the date in the same format as the creation date of the file on diskette.

You cannot specify a date if:

- You delete a library or folder from disk (you specify LIBR or FOLDER).
- You specify ALL.

#### **Notes:**

1. If you do not specify a date and more than one file with the specified name exists on the disk, a message is issued and the operator can delete all the files with that name or cancel the job.
2. If you do not specify a date and more than one file with the specified name exists on a diskette, only the first file (in terms of location) with the specified name is deleted. The physical placement of the VTOC entry for the file on the diskette determines the first file. Determine the placement of the files on diskette by using the CATALOG procedure to list the entries by location.

### **LIBR**

This specifies the deletion of a library from disk. If you do not specify LIBR or FOLDER, a file is deleted from disk. Specify LIBR only if you also specify F1.

You can only delete objects that are not in use or that you are authorized to delete. If you are not authorized to delete one or more of the objects in a library or if one or more of the objects in the library is in use, you cannot delete the library. A message is issued indicating the deletion was incomplete.

### **FOLDER**

This specifies the deletion of a folder from disk. If you do not specify LIBR or FOLDER, a file is deleted from disk. Specify FOLDER only if you specify F1.

If you are not authorized to delete one or more of the objects in a folder or if one or more of the folder's objects is in use, you cannot delete the folder. You can only delete objects that are not in use or that you are authorized to delete. A message is issued indicating the deletion was incomplete.

### **ALL**

This specifies the deletion from disk of files, libraries, and folders that are members of the specified file group. Specify ALL only if you specify F1.

#### **Notes:**

1. If you are not authorized to delete one or more of the objects in a library or if one or more of the objects in the library is in use, you can only delete objects that are not in use or that you are authorized to delete. You cannot delete the library. A message is issued when the deletion is incomplete.
2. If you are not authorized to delete one or more of the objects in a folder or if one or more of the folder's objects is in use, you can only delete objects that are not in use or that you are authorized to

delete. You cannot delete the folder. A message is issued when the deletion is incomplete.

3. If you specify a name for parameter 1 and do not specify LIBR, FOLDER, or ALL for parameter 5, a file is deleted. If there is no file with the specified name, but a library or folder with the name is found, a message is issued. You can then continue with the delete of the library or folder, or you can cancel the job.

#### **starting location and ending location**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

Valid values for the starting and ending locations are:

- S1, S2, or S3
- M1 or M2
- M1.nn or M2.nn (nn is a magazine slot number from 1 to 10)

#### **volume id**

This specifies the volume ID of diskettes to be processed. Use the volume ID to ensure that the correct diskettes are processed. Enter volume ID if the first parameter is ALL and the second parameter is I1.

If the specified volume ID matches the volume ID of the diskettes, the files are deleted. If the specified volume ID does not match the volume ID of the diskettes, a message displays and the operator can either insert the correct diskette or cancel the procedure.

#### **group name**

This specifies the file group to delete from disk. For example, to delete files AB.F1 and AB.F2, which make up file group AB, enter AB without the period. The first parameter must be ALL. The second parameter must be F1.

#### **Example 1**

The following example deletes the file named PAYROLL from the diskette:

```
DELETE PAYROLL,,REMOVE
```

#### **Example 2**

The following example deletes the library named MYLIB from disk:

```
DELETE MYLIB,F1,,,LIBR
```

#### **Example 3**

The following example deletes the folder named MYFLDR from disk:

```
DELETE MYFLDR,F1,,,FOLDER
```

#### **Example 4**

The following example deletes all the files on a diskette with a volume ID of VOL001:

```
DELETE ALL,I1,,,,S1,,VOL001
```

**Example 5**

The following example deletes all the files in the file group named FIL from disk. The files in this group are named FIL.A, FIL.B, and FIL.C.

```
DELETE ALL,F1,,,,,,FIL
```

**Example 6**

The following example removes the PAYROLL file from diskette, and erases all the data that was contained in the file:

```
DELETE PAYROLL,,ERASE
```

---

**DELNRD Procedure**

Not supported. On System/36, the DELNRD procedure removed the network resource directory (#NRD.FLE) from disk.

On the AS/400 system, the information that was stored in the NRD is kept in OS/400 distributed data management (DDM) files. Each DDM file is the equivalent of one NRD entry.

To do a similar function, use the Work with Distributed Data Management Files (WRKDDMF) command or the Delete File (DLTF) command to delete DDM files.

See the *CL Reference* book for more information on the WRKDDMF and DLTF commands. See the *Distributed Data Management* book for more information on using DDM files.

---

**DFA Procedure**

Not supported. On System/36, the DFA procedure retrieved, formatted, and displayed or printed selected information from a dump file.

On the AS/400 system, this function is not required. System dump information is formatted and stored as a spool file. You do not have to request that the dump information be formatted.

---

**DFU Procedure**

The DFU procedure displays a menu from which you can select an option to run an existing data file utility (DFU) program, to create or change a DFU program, to delete a DFU program, or to create and run a temporary DFU program. Using the DFU program, you can update or display data in a disk file. For more information about DFU, see the *ADTS/400: Data File Utility* book. To create a disk file using DFU, see "ENTER Procedure" on page 4-89. To list a data file, see "LIST Procedure" on page 4-134.



S9020106-0

**Example**

The following example displays the DFU menu:

DFU

---

**DFULOAD Procedure**

Not supported. On System/36, the DFULOAD procedure installed the data file utility (DFU) support from diskette. See “Managing Licensed Programs” on page 4-2 for information on installing licensed programs.

---

**DFUSAVE Procedure**

Not supported. On System/36, the DFUSAVE procedure saved the data file utility (DFU) support to diskette. See “Managing Licensed Programs” on page 4-2 for information on saving licensed programs.

---

**DICTLOAD Procedure**

Not supported. On System/36, the DICTLOAD procedure installed the dictionary support for DisplayWrite/36\* (DW/36) from diskette. See “Managing Licensed Programs” on page 4-2 for information on installing licensed programs.

---

**DICTSAVE Procedure**

Not supported. On System/36, the DICTSAVE procedure saved the dictionary support for DW/36 to diskette. See “Managing Licensed Programs” on page 4-2 for information on saving licensed programs.

---

**DISABLE Procedure**

Not supported. On System/36, the DISABLE procedure stopped or disabled an enabled Interactive Communications Feature (SSP-ICF) subsystem, MSRJE, 3270 device emulation, or PC Support/36.

To do similar function, use the Vary Configuration (VRYCFG) command. The VRYCFG command varies AS/400 configuration objects on or off. You can also use the Work with Configuration Status (WRKCFGSTS) command specifying \*LIN for the type (CFGTYPE) parameter.

On System/36, the DISABLE procedure did not remove the auto-monitoring function on an SSP-ICF BSCSEL multipoint tributary line. Therefore, even after running DISABLE, the host system continued to poll a station address associated with that line. On the AS/400 system, there is no similar support. If the device description object is varied off by the VRYCFG command, the location address for that device is no longer polled by the host system.

See the *CL Reference* book for more information on the VRYCFG and WRKCFGSTS commands. See the *Local Device Configuration* book for more information on communications and device configuration.

---

## DISPLAY Procedure

The DISPLAY procedure is supported only for compatibility with the IBM System/34. See "LISTDATA Procedure" on page 4-137 for information on how to display or print the contents of a disk file.

When printing the contents of a file on a double-byte character set version of the OS/400 operating system, the system will determine whether the file may contain double-byte character set data. The DISPLAY procedure cannot display double-byte character set data at a display station.

When you use the DISPLAY procedure to select a range of records for display to your CRT, a temporary intermediate file and library are created.

---

## DLSLOAD Procedure

Not supported. On System/36, the DLSLOAD procedure installed the document library services (DLS) support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## DLSSAVE Procedure

Not supported. On System/36, the DLSSAVE procedure saved the document library services (DLS) support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## DOCCNV Procedure

Not supported. On System/36, the DOCCNV procedure converted all document folders and mail folders, created prior to release 5, to a new internal format.

On the AS/400 system, this function is not required. Folders must be converted before being migrated to the AS/400 system.

See the *Using OfficeVision/400* book and the *Using OfficeVision/400 Word Processing* book for more information on Systems Application Architecture\* (SAA\*) OfficeVision for OS/400\*. See the *Planning for and Setting Up OfficeVision/400* book for more information on folders and documents.

---

## DOCPLOAD Procedure

Not supported. On System/36, the DOCPLOAD procedure installed the print online support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## DOCPSAVE Procedure

Not supported. On System/36, the DOCPSAVE procedure saved the print online support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## DSPLOCKS Procedure

The DSPLOCKS procedure is supported only for compatibility with the IBM System/36. Only the procedure interface is supported. The displays and functions provided by the System/36 environment DSPLOCKS procedure are similar, but not identical, to those provided by the System/36 procedure.

The DSPLOCKS procedure shows the prompt for the Work with Object Locks (WRKOBJLCK) command. On this prompt, you must supply at least the name and the object type of the object for which locks are to be displayed. The WRKOBJLCK command provides the displays that are shown and processes the options that you choose. See the WRKOBJLCK command in the *CL Reference* book for information on how to work with locks on files, libraries, folders, and other objects. See the Display Record Locks (DSPRCDLCK) command in the *CL Reference* book for information on how to display record locks.

Only the locks on one object at a time can be displayed. Record locks are not shown for files.

The DSPLOCKS procedure cannot be used to determine if there are alternative index files associated with a physical file.

---

## DSPMNU Procedure

Not supported. On the System/36, the DSPMNU procedure allows a user to display the text of the commands that are run by the options of a specified menu. There is no similar function available on the AS/400 system.

---

## DSPSYS Procedure

The DSPSYS procedure is supported only for compatibility with the IBM System/36. Only the procedure interface is supported. The displays and functions provided by the System/36 environment DSPSYS procedure are similar, but not identical, to those provided by the System/36 procedure.

The DSPSYS procedure runs the Work with System Status (WRKSYSSTS) command, using the assistance level specified in your user profile. The WRKSYSSTS command provides the displays that are shown and processes the options that you choose. See the WRKSYSSTS command in the *CL Reference* book for information on how to display and work with system status information. See the Work with Hardware Products (WRKHDWPRD) command in the *CL Reference* book for information on how to display and work with the system hardware configuration.

---

## DSU Procedure

The DSU procedure allows you to create, edit, remove, view, or print library procedure members and library source members using the Source Entry Utility (SEU). SEU is a full-screen editor that allows you to edit an entire screen of text or data at a time.

SEU is part of the Application Development Tools licensed program. For more information about SEU, see the *ADTS/400: Source Entry Utility* book.

To create or edit a procedure or source member using the DSU procedure:

```
DSU [EDIT],[member name],[type],[format member],[statement length],
    [library name],[diagnosed source file],[display size]
```

59020505-1

To create, edit, view, print, or remove a library member using the DSU procedure:

```
DSU [LIBRARY],[library name],[type],[format member],,,, [display size]
```

59020506-1

**EDIT** This specifies that an edit session is initialized for the specified member. If the member does not exist, it is created. If you do not specify a member name, the Work with Library Members Using SEU display is shown.

**LIBRARY** This takes you to the Work with Library Members Using SEU display which lists the members in the specified or default library. From there you can create, edit, view, print, or remove a library member.

**member name**

This specifies the source member or procedure that you want to create or edit. A member name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except blanks. Avoid using the following characters because these have special meanings in procedures: commas (,), hyphens (-), apostrophes ('), question marks (?), slashes (/), greater than signs (>), plus signs (+), equal signs (=), or periods (.). If you do not specify a name, a list of the source or procedure members in the library is displayed.

**library name**

This specifies the library that contains or will contain the source or procedure member being changed or created. If you do not specify a library name, the current library is assumed.

**type**

This specifies the type of member to list, create, or edit.

If you specify LIBRARY as the first parameter, P and S are valid entries. Entering a P in this field indicates that you want to list procedure members (all members of the file QS36PRC are displayed). Entering an S in this field indicates that you want to list source members (all members of the file QS36SRC are displayed). If you do not specify a type, S is assumed.

If you specify EDIT as the first parameter, the following entries are valid:

- A** specifies a library source member that contains RPG II auto report specifications.
- F** specifies a library source member that contains SFGR display format S, H, and D specifications.
- P** specifies a library procedure member.

**R** specifies a library source member that contains an RPG II program.

**S** specifies a library source member.

**T** specifies a message source member.

**W** specifies a source member containing Work Station Utility (WSU) statements.

If you do not specify a type, S is assumed.

**format member**

This is supported for System/36 compatibility only. The value is not used.

**statement length**

This is supported for System/36 compatibility only. The value is not used.

**diagnosed source file**

This is supported for System/36 compatibility only. The value is not used.

**display size**

This is supported for System/36 compatibility only. The value is not used.

You can use the SEU Edit Services display to specify the display size as either 24 rows and 80 columns (24 x 80) or 27 rows and 132 columns (27 x 132).

**Example 1**

The following example lets you create or edit source member ACCTRECV in library ACCTLIB:

```
DSU EDIT,ACCTRECV,S,,ACCTLIB
```

**Example 2**

The following example lists the names of all procedure members in library ACCTLIB on your display station. You can then choose to edit, view, print, or remove them.

```
DSU LIBRARY,ACCTLIB,P
```

---

## DSULOAD Procedure

Not supported. On System/36, the DSULOAD procedure installed the development support utility (DSU) support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## DSUSAVE Procedure

Not supported. On System/36, the DSUSAVE procedure saved the development support utility (DSU) support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## DUMP Procedure

Not supported. On System/36, the DUMP procedure allowed you to display or print a specified area on the disk or from a diskette or tape previously created by the APAR procedure. DUMP could also print or display selected sectors from the disk or diskette, and print selected blocks from a tape or tape cartridge.

There is no similar function available on the AS/400 system. The way the AS/400 system manages disk storage is different than the System/36. You can dump job information by using the Dump Job (DMPJOB) and Dump Job Internal (DMPJOBINT) commands. You can also dump the disk storage for an object by using the Dump Object (DMPOBJ) or Dump System Object (DMPSYSOBJ) command. You can use the Dump Tape (DMPTAP) command to dump information from a tape. These commands can be selected from the Dump Commands (CMDDMP) menu.

See the *CL Reference* book for more information on the DMPJOB, DMPJOBINT, DMPOBJ, DMPSYSOBJ, and DMPTAP commands. See the *System Operation* book for more information on working with system problems.

---

## EDITNRD Procedure

Not supported. On System/36, the EDITNRD procedure allowed you to create and edit the network resource directory (#NRD.FLE). When editing the directory, you could add, change, scan, delete, and print the entries that made up the directory.

On the AS/400 system, the information that was stored in the NRD is kept in OS/400 distributed data management (DDM) files. Each DDM file is the equivalent of one NRD entry.

To do similar function, use the Work with Distributed Data Management Files (WRKDDMF) command.

See the *CL Reference* book for more information on the WRKDDMF command. See the *Distributed Data Management* book for more information on using DDM files.

---

## EM3270 Procedure

The EM3270 procedure signs a display station on or printer on or off BSC 3270 emulation. Before you can enter this procedure, the BSC line, controller, and device must be varied on.

To set System/36 environment 3270 Device Emulation values for location name and national language, you must use the Change System/36 (CHGS36) command.

See the *3270 Device Emulation Support* book for more information about emulation and its command interface.

To sign on a display station:

EM3270	[ display id ]	,	[ ON OFF ]	,	[ location name ]	,	[ display messages ]
							[ Y N ]

59020111-1

If you are signing on your own display station, you need to enter only EM3270.

### display id

This specifies the name of the display station to sign on BSC 3270 emulation. If the display station being signed on is the one from which the procedure is entered, do not use this parameter. If the display ID is that of another display station, that display station must be a data display station that is not currently in use and the procedure must be run in a batch environment.

**ON** This specifies that you want to sign a display station on to BSC 3270 device emulation. If you do not enter a parameter, ON is assumed.

**OFF** This specifies that you want to sign a display station off BSC 3270 emulation. This value is ignored and ON is always assumed.

### location name

This specifies the remote location name associated with this session. The remote location name is defined when creating a BSC device (CRTDEV BSC command) for 3270 emulation.

### display messages

This is supported for System/36 compatibility only. The value is not used.

**YES or Y** This is supported for System/36 compatibility only. The value is not used.

**NO or N** This is supported for System/36 compatibility only. The value is not used.

To sign on or off a printer:

```
EM3270 [printer id], [ON
        OFF], [location name], [spooling],
        [defer printing], [priority]
```

S9020112-1

**printer id** This specifies the printer device name of the printer to sign on or off. If the printer output for this printer is not being spooled, the printer must not be in use.

**ON** This specifies that you want to sign a printer on to BSC 3270 emulation. If you do not enter a parameter, ON is assumed.

**OFF** This specifies that you want to sign a printer off BSC 3270 device emulation.

**location name**

This specifies the location name associated with this session. The location name is defined when creating a BSC device (CRTDEV/BSC command) for 3270 emulation and refers to the remote location with which communications takes place.

**spooling** This specifies whether to spool output to the printer. Specify either YES or NO.

**YES or Y** This specifies the spooling of output. If you do not specify a parameter, YES is assumed.

**NO or N** This specifies that output is not spooled.

**defer printing**

This specifies when spooled output prints. Specify either YES or NO. If spooling is not active, the parameter is ignored.

**YES or Y** This indicates that the printing of spooled output waits until the printer is signed off BSC 3270 emulation. If you do not enter a parameter, YES is assumed.

**NO or N** This indicates that printing of the spooled output does not wait until the printer is signed off 3270 emulation. That is, print the output as soon as it can print.

**priority** This is supported for System/36 compatibility only. The value is not used.

**Note:** The following examples assume that a default emulation location name and national language has been set changing the System/36 environment configuration (CHGS36 command).

**Example 1**

The following example signs your display station on to 3270 emulation:

```
EM3270
```

## Example 2

The following example signs printer P3 on to 3270 emulation. The printed output spools, and any output prints as soon as possible.

```
EM3270 P3,ON,,,NO
```

---

## ENABLE Procedure

Not supported. On System/36, the ENABLE procedure started or enabled an Interactive Communications Feature (SSP-ICF) subsystem, MSRJE, 3270 device emulation, or PC Support/36.

To do similar function, use the Vary Configuration (VRYCFG) command. The VRYCFG command varies AS/400 configuration objects on or off. You can also use the Work with Configuration Status (WRKCFGSTS) command specifying \*LIN for the type (CFGTYPE) parameter.

See the *CL Reference* book for more information on the VRYCFG and WRKCFGSTS commands. See the *Local Device Configuration* book for more information on communications and device configuration.

---

## ENTER Procedure

The ENTER procedure allows you to create or change a data file utility (DFU) program so you can use this utility to create a disk file. For more information about DFU, see the *ADTS/400: Data File Utility* book.

```
ENTER  file name,dfu program name, [ file source member name ],
      number of records, [ D ] , [ NN ] , [ dfu source member name ] , ,
                        [ Z ] , [ NY ]
                        [ B ] , [ YN ]
                               [ YY ]
                               [ GO ]

[ library name ] , [ display source member name ] , [ name of file on disk ]
  current library
```

59020114-0

**file name** This specifies the file to create. The file name can be from 1 through 8 alphameric characters.

### dfu program name

This specifies the DFU program used to process the file. If the program does not exist in the library, DFU starts the setup procedures to create the program. If the program exists in the library specified, DFU runs it. To change an existing DFU program, leave this name blank and specify the name of the program to be changed in the DFU source member name parameter. If both the DFU program name and the DFU source member name are not provided, DFU starts the setup procedures to create a temporary program.

See "ENTER# Procedure" on page 4-91 for alternate methods of running a DFU program.

**file source member name**

This specifies the RPG II source member that contains the file description (F-specification) and record input descriptions (I-specifications) that describe the file to process. This member can contain one or more sets of file description and input specifications, or an entire RPG II program. The file description and input specifications that correspond to the file are taken as the data description.

You receive a prompt for this parameter if you do not specify it. This parameter is required.

**number of records**

This specifies the maximum number of records you want to enter into the file at file creation. This parameter is required. If you do not specify the parameter, you receive a prompt for it.

**D, Z, or B**

This indicates whether unkeyed selected zoned numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are D, Z, or B. If no parameter is specified, D is assumed.

**D or B** This specifies a data file with blank fill of unkeyed selected zoned numeric fields.

**Z** This specifies a data file with zero fill of unkeyed selected zoned numeric fields.

**NN, NY, YN, YY, or GO**

This is supported for System/36 compatibility only. The value is not used.

**DFU source member name**

To change an existing DFU program, specify the name of the DFU program in this parameter and leave the DFU program name parameter blank. If the DFU program is specified, this parameter is ignored.

**library name**

This specifies the library containing or to contain the DFU program. All library members associated with the DFU job are looked for, or stored, in this library. If you do not specify this parameter, the current library is assumed.

**display source member name**

This parameter is not supported. If a value is specified, an error message will be issued.

**name of file on disk**

This specifies the name in the disk VTOC for the file to create, if it is different from the name specified in the DFU program. If you specify the name of a file on disk and fail to specify a file to be created by the DFU program, you are prompted for that parameter. This parameter is also referred to as FILE LABEL.

## Example

The following example allows you to create a DFU program called DFILFMT to create a disk file named FILE1 using a file source member named RPGPROG. The file contains 100 records. The library MYLIB contains the file source member and will contain the DFU specifications.

```
ENTER FILE1,DFILFMT,RPGPROG,100,,,,,MYLIB
```

---

## ENTER# Procedure

The ENTER# procedure allows you to create a disk file using the data file utility (DFU) by running the specified program. For more information about DFU, see the *ADTS/400: Data File Utility* book.

```
ENTER#  file name,dfu program name, [ file source member name ],
        number of records, [ D
                           Z
                           B ], [ NN
                                 NY
                                 YN
                                 YY
                                 GO ], [ dfu source member name ],,
        [ library name
          current library ], [ display source member name ], [ name of file on disk ]
```

59020654-0

**file name** This specifies the file to create. The file name can be from 1 through 8 alphanumeric characters.

### DFU program name

This specifies the DFU program to run to create the file. The program must already exist in the specified library.

### file source member name

This is supported for System/36 compatibility only. The value is not used.

### number of records

This specifies the maximum number of records you want to enter into the file at file creation. This parameter is required. If you do not specify the parameter, you receive a prompt for it.

### D, Z, or B

This indicates whether unkeyed selected zoned numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are D, Z, or B. If no parameter is specified, D is assumed.

**D or B** This specifies a data file with blank fill of unkeyed selected zoned numeric fields.

**Z** This specifies a data file with zero fill of unkeyed selected zoned numeric fields.

### NN, NY, YN, YY, or GO

This is supported for System/36 compatibility only. The value is not used.

### DFU source member name

This is supported for System/36 compatibility only. The value is not used.

**library name**

This specifies the library containing or to contain the DFU program. All library members associated with the DFU job are looked for, or stored, in this library. If you do not specify this parameter, the current library is assumed.

**display source member name**

This is supported for System/36 compatibility only. The value is not used.

**name of file on disk**

This specifies the name in the disk VTOC for the file to create, if it is different from the name specified in the DFU program. If you specify the name of a file on disk and fail to specify a file to be created by the DFU program, you are prompted for that parameter. This parameter is also referred to as FILE LABEL.

**Example**

The following example creates a disk file named FILE1 using a DFU program named DFILFMT. The file contains 100 records. The library MYLIB contains the DFU program.

```
ENTER# FILE1,DFILFMT,,100,,,,MYLIB
```

---

**EPDOWNL Procedure**

Not supported. On System/36, the EPDOWNL procedure copied the personal computer portion of the 3278 emulation via IBM Personal Computer from the System/36 to the IBM Personal Computer.

To do similar function, run 3270 display emulation using "ES3270 Procedure" on page 4-95 or the Start 3270 Display Emulation (STREML3270) command.

---

**EPLMRG Procedure**

Not supported. On System/36, the EPLMRG procedure merged the personal computer machine-readable instruction (MRI) files and translated tables in #LIBRARY into the virtual diskette #EPPCLD1.

There is no similar function available on the AS/400 system.

---

**EP3270 Procedure**

The EP3270 procedure signs a display station on to SNA 3270 display emulation. Before you can enter this procedure, the host line, controller, and device must be varied on. To set System/36 environment 3270 device emulation values for location name and national language, you must use the Change System/36 (CHGS36) command.

See the *3270 Device Emulation Support* book for more information about 3270 emulation and its command interface.

```
EP3270 [ location name ]
```

59020507-0

**location name**

This specifies the remote location associated with this session. The remote location name is defined when creating a host device (CRTDEVHOST command) for 3270 emulation and refers to the name of the remote location with which communications takes place. If a default emulation location name was defined in the System/36 environment configuration, you can omit this parameter.

**Note:** The following example assumes that a default emulation location name has been set by changing the System/36 environment configuration (CHGS36 command).

**Example**

The following example signs your display device on to 3270 device emulation:

```
EP3270
```

**ERAP Procedure**

Not supported. On System/36, the ERAP procedure displayed or printed data that was logged for the devices on the system.

There is no similar function available on the AS/400 system. You can find out about errors that have occurred on the system by using the Work with Problems (WRKPRB) command. You can print the system error log entries by using the Print Error Log (PRTERLOG) command or by running the *Work with error log* service function provided by the Start System Service Tools (STRSST) command.

See the *CL Reference* book for more information on the WRKPRB, PRTERLOG, and STRSST commands. See the *System Operation* book for more information on working with system problems. See the *AS/400 Licensed Internal Code Diagnostic Aids – Volume 1* book for more information on the system service tools (SST).

**ERR Procedure**

The ERR procedure causes the specified error message to display, and allows an operator to select an option: 0, 1, 2, 3, or D. The specified message is retrieved from the current user message member (the OS/400 message file). This allows your procedures to display messages and issue options in the same way that the system displays error messages.

For information on assigning the USER1 message member, see “MEMBER OCL Statement” on page 5-59.

The ERR procedure runs the \$CPPE utility program.

```
ERR message id, [ options ], [ 'text' ]
```

59020115-0

**message id code**

This specifies the message identification code of the displayed message. The parameter is a 1- to 4-digit number in the range 0 through 9999.

**options** This specifies the options to display with the message. The options can be any combination of 0, 1, 2, or 3, but you must enter the numbers in ascending order and with no blanks or other characters. For example, to cause options 1 and 3 to be displayed, enter 13. If you specify no options, option 3 is assumed.

If an operator selects option 3, the procedure is immediately canceled. Control is not returned to the procedure. If an operator selects option 0, 1, or 2, a return code is set. Use the ?CD? expression in a procedure to check for the option selected by an operator.

The value set is as follows:

	<b>?CD? Return</b>
<b>Option</b>	<b>Code Value</b>
<b>0</b>	1010
<b>1</b>	1011
<b>2</b>	1012

See "Substitution Expressions" on page 3-7 for more information about substitution expressions.

If an operator selects option D, a job process dump is taken and the procedure immediately cancels. The process level dump provides the same information as is provided by the Dump Job (DMPJOB) command.

**text** This specifies the text to be used as replacement characters in the message. These will be inserted into the & fields of the message in a left to right order. Attempting to insert into a & field that is not defined as a CHAR field may result in an error. This field may be up to 75 bytes long and must be enclosed in apostrophes. No embedded apostrophes are allowed.

**Example**

The following example shows a procedure that displays an error message from the MESSAGES message member if the operator did not enter SOURCE or PROC for the second parameter. That is, parameter 2 does not equal SOURCE or PROC. The message displayed is number 0020 from the level-one message member named MESSAGES. The allowed options are 0, 1, and 3:

- If option 0 is entered, parameter 2 (P2) contains SOURCE.
- If option 1 is entered, parameter 2 (P2) contains PROC.
- If option 3 is entered, the procedure is automatically canceled.

```
// MEMBER USER1-MESSAGES
// IFF ??=SOURCE IFF ??=PROC ERR 0020,013
// IF ?CD?=1010 EVALUATE P2='SOURCE'
// IF ?CD?=1011 EVALUATE P2='PROC'
LISTLIBR ?1?,??,MYLIB
```

If the operator did not enter a second parameter, message 0020 can display as follows:

```
USR0020 (01 3)
SOURCE or PROC must be entered. 0=SOURCE, 1=PROC, 3=Cancel
```

The source statements contained in the MESSAGES member are:

```
MESSAGES,1
0020 SOURCE or PROC must be entered. 0=SOURCE, 1=PROC, 3=Cancel
```

---

## ES3270 Procedure

The ES3270 procedure signs a display station on or printer on or off SNA 3270 emulation. Before you can enter this procedure, the host line, controller, and device must be varied on. To set System/36 environment 3270 device emulation values for location name and national language, you must use the Change System/36 (CHGS36) command.

See the *3270 Device Emulation Support* book for more information about 3270 emulation and its command interface.

To sign on a display station:

```
ES3270 [display id], [ON
OFF], [location name], [printer id]
```

SP020116-0

### display id

This specifies the name of the display station to sign on SNA 3270 emulation. If the display station being signed on is the one from which the operator enters the procedure, this parameter is not required. If the display ID is that of another display station, that display station must be a data display station that is not currently in use and the procedure must be run in a batch environment.

**ON** This specifies that you want to sign a display station on to SNA 3270 emulation. If you do not enter a parameter, ON is assumed.

**OFF** This specifies that you want to sign a display station off SNA 3270 emulation. This value is ignored and ON is always assumed.

### location name

This specifies the remote location name associated with this session.

The remote location name is defined when creating a host device (CRTDEVHOST command) for 3270 emulation and refers to the name of the remote location with which communication takes place. If a default emulation location name was defined in the System/36 environment configuration, you need not specify this parameter.

**printer id** This is supported for System/36 compatibility only. The value is not used.

To sign on or off a printer or close a spool file:

```

ES3270 [ printer id ], [ ON
          OFF
          CLO ], [ location name ], [ spooling
          Y
          N ], [ defer printing
          Y
          N ],

[ priority
  1 ], [ positions
  132 ], [ lines per page ], [ OP
  BR
  nn ]

```

S9020117-2

**printer id** This specifies the printer device name of the printer to sign on or off SNA 3270 emulation. If the printer output for this printer is not being spooled, the printer must not be in use.

**ON** This specifies that you want to sign a printer on to SNA 3270 emulation. If you do not enter a parameter, ON is assumed.

**OFF** This specifies that you want to sign a printer off SNA 3270 device emulation.

**CLO** This specifies the closing of the current spooled file and another one opened in order to continue with the SNA 3270 printer emulation operation.

#### location name

This specifies the remote location name associated with this session. The remote location name is defined when creating a host device (CRTDEVHOST command) for 3270 emulation and refers to the name of the location with which communication takes place. If a default emulation location name was defined in the System/36 environment configuration, you need not specify this parameter.

**spooling** This specifies whether output to the printer is spooled. Specify either YES or NO.

**YES** This specifies that output is spooled. If you do not specify a parameter, YES is assumed.

**NO** This specifies that output is not spooled.

#### defer printing

This specifies when to print spooled output. Enter either YES (Y) or NO (N). If spooling is not active, this parameter is ignored. YES indicates that the printing of spooled output waits until the printer is signed off SNA 3270. NO indicates that printing of the spooled output does not wait until the printer is signed off SNA 3270. That is, output prints as soon as output can print. If you do not enter a parameter, YES is assumed.

**priority** This is supported for System/36 compatibility only. The value is not used.

**positions** This specifies the number of print positions in a line. Specify any decimal number from 1 through 132. If you do not specify a parameter, 132 is assumed.

#### lines per page

This specifies the number of lines per page. Specify any decimal number from 1 through 255. If you do not enter a parameter, the current number of lines per page is assumed.

- NO** This specifies that the operator controls the spool file close.
- BR** This specifies the creation of spooled files for each group of data received in brackets from the host.
- nn** This specifies an inactive time value in minutes. Values can be from 01 to 99. When the host system has not sent data for the specified time, the spooled file closes. If a spooled file is closed, another spooled file opens with the same inactive conditions.

**Note:** The following examples assume a default emulation location name has been set. Use the Change System/36 (CHGS36) command to set the default emulation location name.

### Example 1

The following example signs your display station on to 3270 device emulation:

```
ES3270
```

### Example 2

The following example signs printer P3 on to 3270 device emulation. The printed output is spooled, and any output prints as soon as possible.

```
ES3270 P3,ON,,,NO
```

---

## EXTRACT Procedure

The EXTRACT procedure is supported only for compatibility with the IBM System/34. (System/36 does not allow you to specify hexadecimal data in a procedure, however, so the seventh parameter in the EXTRACT procedure cannot contain hexadecimal data.)

The EXTRACT procedure can also be used with physical, database data files that do not have a System/36 environment direct, indexed, or sequential file organization. When such files are used, the new file may be created with file attributes that differ from those you expect. A message allowing you to continue is sent whenever this may occur. See the *System/36 Environment Programming* book for more information on using System/36 environment files and AS/400 files.

If you do not supply a position value for the sixth parameter, and INCLUDE and EQ have been specified for the fourth and fifth parameters, every position in each record will be compared with the character string specified in the seventh parameter.

When you use the EXTRACT procedure to display the contents of a disk file on your display station and INCLUDE or OMIT processing is performed, or a limited number of records are copied, a temporary intermediate disk file and library are created. See "COPYDATA Procedure" on page 4-50 for information about copying data files. See "LISTDATA Procedure" on page 4-137 for information about listing data files.

---

## FLIB Procedure

Use the FLIB procedure to specify the following:

- The session files library that is to be used to search for database files in the System/36 environment.
- The library list search indicator. The library list search indicator determines whether the library list should be used to locate database files in the System/36 environment.

The current files library is initially set from the session files library for each job. Similarly, the current library list search indicator is initially set from the session library list search indicator for each job. When you are entering procedures, control commands, or OCL statements from the keyboard, the session files library and the current files library are the same. Similarly, the session library list search indicator and the current library list search indicator are also the same.

When a file is created in the System/36 environment, it is created in the current files library. Files used within the System/36 environment are found by searching the current files library. However, if the current library list search indicator is set to search the library list, the system searches the library list for the specified files.

You can change the current files library and the current library list search indicator for the current procedure by using the FILELIB OCL statement. See "FILELIB OCL Statement" on page 5-43 for more information. If you change the current files library or the current library list search indicator, the session files library does not change.

The specified library remains the session files library and the specified library list search indicator remains the session library list search indicator until you do one of the following:

- Sign off the system.
- Leave the System/36 environment by specifying either the ENDS36 or the TFRJOB command.
- Enter another FLIB procedure to change the session files library or session library list search indicator.
- Enter a FILELIB OCL statement to change the session files library or session library list search indicator.
- Use a FILELIB OCL statement with SESSION-YES to change the session files library or session library list search indicator within a procedure.

The FLIB procedure processes a FILELIB OCL statement with the SESSION-YES specified. See the *System/36 Environment Programming* book for more information about changing the files library or searching the library list.

The following is the syntax diagram for the FLIB procedure:

```

FLIB [library name] [,LIBL
                                ,NOLIBL]

```

59020688-2

### library name

This specifies the library name to be used as the session files library.

**LIBL** This specifies that the session library list search indicator is set so that a search is performed on the library list for database files. If you specify this type of search, the current files library is not searched unless the current files library is in the library list.

**NOLIBL** This specifies that the session library list search indicator is set so that a search is not performed on the library list for database files. If you specify this type of search, the current files library is the only library that is searched for database files.

**Note:** At least one parameter must be specified on the FLIB procedure. If you do not specify a library name, the session files library does not change. If you do not to specify either the LIBL or the NOLIBL parameter, the session library list search indicator will not change.

### Example 1

The following example changes the session files library to a library called MYLIB. By changing the session files library, the session library list search indicator does not change.

```
FLIB MYLIB
```

### Example 2

The following example changes the session files library to a library called MYLIB. By specifying the LIBL parameter, the session library list search indicator is set so that the library list is searched for all database files.

```
FLIB MYLIB,LIBL
```

### Example 3

The following example sets the session library list search indicator so that a search of the library list for all database files does not occur. By specifying the NOLIBL parameter, a search of the current files library for all database files occurs. The session files library does not change.

```
FLIB ,NOLIBL
```

---

## FORMAT Procedure

The FORMAT procedure processes source statements you have created called **display format** specifications or SFGR source. Display format specifications define how information is displayed at a display station. Use the source entry utility (SEU) or screen design aid (SDA) to create the display format specifications. When you use SDA, you need not use the FORMAT procedure. The FORMAT procedure is equivalent to the Create System/36 Display File (CRTS36DSPF) command.

See Appendix D for more information on error and warning messages produced by the FORMAT procedure when syntax checking of the SFGR source specifications.

For information about display format specifications see the *System/36 Environment Programming* book. For sample listings of the FORMAT procedure, see the *ADTS/400: Screen Design Aid for the System/36 Environment* book.

The FORMAT procedure does the following:

- Creates a new display format load member (the OS/400 display file) containing the formats defined by the source specifications. You can place all display

## FORMAT

formats used by a program in one or more display format load members. Place up to 255 display formats in one member.

- Adds one or more display formats to an existing display format load member.
- Replaces one or more display formats in an existing display format load member.
- Deletes a display format from an existing display format load member.

When you generate a display format load member, the reference number assigned to the source member is assigned to the generated load member. This allows you to determine whether an old level of a load member is being used by comparing the two reference numbers. The reference number is located in the first six characters of the object text of the OS/400 display file.

The FORMAT procedure runs the \$SFGR utility program.

To create, add to, or update a display format load member:

```
FORMAT  [CREATE  
        ADD  
        UPDATE] ,load member name, [load member library  
        current library] ,source member name,  
  
        [source member library  
        current library] , [number of formats  
        1] , [REPLACE] , [HALT  
        NOHALT] ,  
  
        [PRINT  
        NOPRINT  
        PARTIAL]
```

59020118-0

To delete a format from a display format load member:

```
FORMAT  DELETE,load member name, [load member library  
        #LIBRARY] ,display format name
```

59020660-0

**CREATE** This specifies the creation of a new display format load member. If you do not specify a parameter, CREATE is assumed.

**ADD** This specifies the addition of one or more display formats to an existing display format load member. Only the display format(s) to add are contained in the source member.

**UPDATE** This specifies the replacement of one or more display formats in an existing display format load member. Only the display format(s) to update are contained in the source member.

**DELETE** This specifies the removal of a single display format from an existing display format load member. If the removed format was the only one in the member, the entire display format load member is removed from the library.

### **load member name**

This specifies the display format load member to create, add to, update, or delete from.

**load member library**

This specifies the name of the library that is to contain (or does contain) the display format load member. If you do not specify a library name, the current library is assumed, unless DELETE is specified. If DELETE is specified, #LIBRARY is the default library.

**source member name**

This specifies the name of the library source member that contains the display format specifications. The source member must be located in a source file named QS36SRC. When you specify DELETE, give the FORMAT name instead of a source member name.

**source member library**

This specifies the name of the library that contains the source member. If you do not specify a library name, the current library is assumed.

**number of formats**

This must be 1 to 255. However, the value is ignored.

**REPLACE**

This specifies that the created display format load member can replace an existing load member with the same name. REPLACE is always assumed when CREATE is not specified, but the existing load member must be an OS/400 display file created by \$SFGR. During a CREATE operation, REPLACE can replace any type of load member, including an OS/400 message file, program, or display file. If the existing load member is not a display file, a message instructs the user to continue or cancel the request.

**Note:** A display file is not created if another type of OS/400 file with the same name exists in the load member library.

**HALT**

This specifies the display of a message indicating that warning and terminating errors were encountered during the processing of the display format source member. If a message displays indicating a warning, the operator can either continue the job or cancel the job. If a message displays indicating a terminating error, the operator can only cancel the job. If you do not enter a parameter, HALT is assumed.

**NOHALT**

This specifies that no message displays indicating that warning or terminating errors were encountered during the processing of a display format source member. For warning errors, the job step is completed and the display formats are generated. For terminating errors, the job step ends and no display formats are generated.

**Note:** For terminating errors, the ?CD? substitution expression is set to 1008. See "Substitution Expressions" on page 3-7 for more information.

**PRINT**

This specifies the printing of the following:

- The display format source member name
- The display format S- and D-specifications
- Any informational, warning, or terminating error messages
- The input and output field descriptions
- A list of the display format indicators used
- The input and output library names
- The display format load member name
- The number of bytes of help text storage

## FORMAT

PRINT can also produce an output listing from the Create Display File (CRTDSPF) command.

If you do not specify a parameter, PRINT is assumed.

**NOPRINT** This specifies that when a terminating error is encountered in the source specifications, only the statement in error and the terminating error message prints. If no errors are found, nothing prints. However, if there are errors, NOPRINT may produce an output listing from the CRTDSPF command.

**PARTIAL** This specifies the printing of the following:

- The display format source member name
- Any warning or terminating messages together with the statement causing the message, or any informational messages
- The input and output library names
- The display format load member name

PARTIAL may also produce an output listing from the CRTDSPF command.

### **display format name**

This specifies the name of the display format to delete.

If the display is used by a MRT program, or if the program acquires multiple work stations for use with the display file, the source member should have a MAXDEV (maximum number of devices) source attribute value (the sum of the number of MRTs and acquired devices used). If the MAXDEV attribute is not set, the default is 5. You can set the MAXDEV source attribute with the Change System/36 Source Attributes (CHGS36SRCA) and Edit System/36 Source Attributes (EDTS36SRCA) commands. Use the Change Display File (CHGDSPF) command to change the MAXDEV value for a display file that is already created.

### **Example 1**

The following example creates a new display format load member named FORMAT2 in the library named MYLIB. The display format specifications define three display formats and are in a source member named INPUT. The source member is in the library named MYLIB.

```
FORMAT CREATE,FORMAT2,MYLIB,INPUT,MYLIB,3
```

### **Example 2**

The following example updates an existing display format load member named FORMAT2 in the library MYLIB. The source member containing the display format specifications is named SCREEN and is also contained in the library MYLIB.

```
FORMAT UPDATE,FORMAT2,MYLIB,SCREEN,MYLIB
```

### **Example 3**

The following example removes a display format named SCREEN1 from the display format load member named FORMAT2. The load member is in the library named MYLIB.

```
FORMAT DELETE,FORMAT2,MYLIB,SCREEN1
```

**Example 4**

The following example adds to an existing display format load member named FORMAT2 in the library MYLIB. The source member containing the display format specifications is named SCREEN and is also contained in the library MYLIB.

```
FORMAT ADD,FORMAT2,MYLIB,SCREEN,MYLIB
```

---

**FORTRAN Procedures**

The following FORTRAN procedures are not supported:

- FORTC
- FORTCG
- FORTGO
- FORTLOAD
- FORTONL
- FORTP
- FORTRANC
- FORTSAVE
- FORTSDA
- FORTSEU

On the AS/400 system, you must write your FORTRAN programs in another high-level language (like COBOL or PL/I).

---

**FROMLIBR Procedure**

The FROMLIBR procedure copies one or more library members to a disk, diskette, tape, or tape cartridge file.

Library members copied to disk, diskette, tape, or tape cartridge files using the FROMLIBR procedure in sector mode (not specifying a record length) cannot be copied to System/32, System/34, or System/36. Use the Save System/36 Library Member (SAVS36LIBM) command to copy source and procedure members you want to restore on System/32, System/34, or System/36. Source and procedure members copied to disk, diskette, tape, or tape cartridge files using FROMLIBR in record mode (specifying a record length) can be copied to System/32, System/34, or System/36.

The LIBRFILEs created by the FROMLIBR procedure are divided into *sector* modes and *record* modes.

The sector mode LIBRFILE is a save file (SAVF) created by the Save Object (SAVOBJ) command. The sector mode LIBRFILE has the following restrictions:

- This file can only be exchanged with another AS/400 system.
- A disk sector mode LIBRFILE can only be copied to a library that has the same name as the library it was copied from.
- The disk sector mode LIBRFILE cannot be processed by a user application program.
- The disk sector mode LIBRFILE cannot be processed by another System/36 utility program because LIBRFILE is not a database file.
- The ADD function is not supported for sector mode LIBRFILE.

## FROMLIBR

The record mode file is a data file created by the Save System/36 Library Member (SAVS36LIBM) command. The record mode LIBRFILE has the restriction that the ADD function is not supported for diskette or tape.

You cannot use the FROMLIBR procedure to copy members from library QSYS.

To save an entire library on diskette, tape, or tape cartridge (even the libraries that contain IBM-supplied members), see "SAVELIBR Procedure" on page 4-228.

The FROMLIBR procedure runs the \$MAINT utility program.

The FROMLIBR procedure creates an AS/400 save file if the record length parameter is not specified.

To copy one or more library members to diskette, tape, or tape cartridge:

```
FROMLIBR { member name  
          member name,ALL } , [ SOURCE  
                              (S)  
                              PROC  
                              (P)  
                              LOAD  
                              (O)  
                              SUBR  
                              (R)  
                              LIBRARY ] , [ file name  
                                           member name ] , [ I1  
                                                           T1  
                                                           T2  
                                                           TC ] , [ retention days ] ,  
  
          volume id , [ library name  
                      current library ] , [ S1  
                                           S2  
                                           S3  
                                           M1.nn  
                                           M2.nn ] , [ AUTO  
                                           NOAUTO ] , [ REWIND  
                                           LEAVE  
                                           UNLOAD ] ,  
  
          [ record length ] , [ SVATTR ]
```

59020128-1

To copy one or more library members to a sequential disk file:

```
FROMLIBR { member name  
          member name,ALL } , [ SOURCE  
                              (S)  
                              PROC  
                              (P)  
                              LOAD  
                              (O)  
                              SUBR  
                              (R)  
                              LIBRARY ] , [ file name  
                                           member name ] , F1 ,  
  
          [ [ T  
            J  
            S ] , [ blocks  
                  g ] ] , [ library name  
                          current library ] , , , , [ record length ] , [ SVATTR ]  
  
          ADD ,
```

59020129-0

### member name

This specifies the library member to copy from the library.

### member name,ALL

This specifies the copying of one or more library members with names beginning with **member name**. The name can be up to 7 characters long. For example: PAY,ALL specifies all library members having names that begin with PAY are copied, such as: PAYROLL, PAY-CHECK, or PAYRUN.

**ALL** This specifies the copying of all library members from the library.

**SOURCE or S**

This specifies that only source members are copied. If you do not specify a parameter, SOURCE is assumed. Source members are all the members in a source file named QS36SRC in the specified library.

**PROC or P**

This specifies that only procedure members are copied. Procedure members are all the members in a source file named QS36PRC in the specified library.

**LOAD or O**

This specifies that load members are copied. Load members are all objects in a library with the following OS/400 object types:

**\*PGM** Programs  
**\*MSGF** Message files  
**\*FILE** Display files (attribute of \*DSPF)

**SUBR or R**

This specifies that only subroutine members are copied. Subroutine members are all objects in a library with the object type of \*PGM (programs).

**LIBRARY** This specifies that all types of members (SOURCE, PROC, LOAD, and SUBR), including all objects with the following OS/400 object types, are copied:

**\*CHTFMT** Chart formats  
**\*CLS** Classes  
**\*CMD** Commands  
**\*DTAARA** Data areas  
**\*FCT** Forms control tables  
**\*FILE** Files (files with all attributes: PF, LF, PRTF, and so on)  
**\*GSS** Graphic symbol sets  
**\*JOB** Job descriptions  
**\*JOBQ** Job queues  
**\*MENU** Menus  
**\*MSGQ** Message queues  
**\*OUTQ** Output queues  
**\*QRYDFN** Query definitions  
**\*SBSD** Subsystem descriptions  
**\*SSND** Session descriptions  
**\*TBL** Tables

**file name** This specifies the file to create. Specify a file name of up to 8 characters in length and begin it with an alphabetic character (A through Z, #, @, or \$). The remaining characters are any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. Do not use ALL as a file name. If you do not specify a file name, the name specified for the member name or partial member name is assumed. If ALL is specified but you do not specify a file name, a message displays and the operator must enter the name of the file to create.

**I1** This specifies the placement of one or more library members in a new diskette file. If you do not specify a parameter, I1 is assumed.

- F1** This specifies the placement of one or more library members in a new disk file.
- T1** This specifies that the tape used for this procedure is on tape drive 1.
- T2** This specifies that the tape used for this procedure is on tape drive 2.
- TC** This specifies that a tape cartridge is used for this procedure.

**retention days**

This specifies how long the diskette, tape, or tape cartridge file is retained, and can be from 0 (zero) through 999 days. If I1, T1, T2, or TC is specified and you do not specify retention days, 1 day is assumed. If you specify a retention period of 999 days, the diskette, tape, or tape cartridge file is a permanent file. For more information on diskette, tape, and tape cartridge file retention, see "FILE OCL Statement (for Diskette Files)" on page 5-33 and "FILE OCL Statement (for Tape Files)" on page 5-37.

- ADD** This specifies the addition of the selected source and procedure members to a record-mode disk file. The disk file must have been created either by a previous record-mode copy, or by using the BLDFILE procedure to create a resident sequential file with a record length between 40 and 120.

When ADD-YES is specified, the RECL parameter must also be specified to indicate that the copy is to be done in record-mode. The record length of the existing file will be used instead of the RECL value.

If ADD-YES is specified when copying members to diskette or tape, or when copying members in sector-mode to a disk file, an error message appears.

**volume id**

This specifies the volume ID of the diskette, tape reel, or tape cartridge. Specify from 1 through 6 alphameric characters. If you do not specify the volume ID, a message displays and the operator must enter the volume ID.

- T** This specifies that the disk file containing the library members is a resident file. If you do not enter a parameter, T is assumed and a new disk file is created. The file remains on disk after the FROMLIBR procedure ends.
- J** This specifies that the disk file containing the library members is a job file. The file does not exist after the job containing the FROMLIBR procedure ends.
- S** This specifies that the disk file containing the library members is a scratch file. The file does not exist after the FROMLIBR procedure ends.

**Note:** For more information on disk file retention (the S, J, and T parameters), see "FILE OCL Statement (for Disk Files)" on page 5-25.

- blocks** This specifies the size of the new disk file to create. One disk block contains 2560 bytes. One byte contains one character. The blocks value is ignored if you specify ADD. If you do not specify blocks, a size of 8 blocks is assumed.

**library name**

This specifies the library containing the one or more library members to copy. If you do not specify a library name, the current library is assumed.

**S1, S2, S3, M1.nn, and M2.nn**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**AUTO**

If specified for tape (unit is T1 or T2) when the tape reel on the original tape drive finishes, the system switches to the other tape drive to continue processing. If the tape reel on the other tape drive is not the same density as the original tape reel, an error message will be issued. If the other tape drive is not available, the system uses the original tape drive.

**NOAUTO**

If specified for tape (unit is T1, T2, or TC) when the tape reel on the original tape drive finishes, the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is tape and neither AUTO nor NOAUTO was specified, AUTO is assumed.

If TC is specified for the unit, the AUTO/NOAUTO parameter is ignored.

If the unit is diskette (11), the AUTO/NOAUTO parameter is supported for compatibility only. If specified, the value will be syntax-checked, but ignored when the diskette is processed.

**REWIND**

If specified for a reel-to-reel tape, the tape is rewound to the load point after processing is complete.

If specified for a tape cartridge, the tape is positioned to the beginning of the cartridge after processing is complete.

**LEAVE**

If specified for a reel-to-reel tape, or a tape cartridge, the tape is left where it was last processed. The next operation to the tape will begin at that point.

**UNLOAD**

If specified for a reel-to-reel tape drive, the tape is rewound and unloaded after processing is complete.

REWIND, LEAVE, or UNLOAD are only valid if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge is processed or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, the preparation is a lengthy process. The preparation time can be shortened if UNLOAD was specified, because the tape is left at the end (not rewound) after the last operation is completed.

**record length**

This specifies the record length, in bytes, of the source or procedure members being copied. Load and subroutine members are not copied. The record length can be from 40 through 120 bytes. Specify this parameter only if you want to create a record mode file.

You can specify the record length parameter only with a member type of SOURCE (S), PROC (P), or LIBRARY. The members are saved in

record mode format. Any members added to a record mode file assume the record length specified for the file. Any other record length specified with the ADD parameter is ignored. Likewise, you must not specify a record length when adding to a file that has no specified record length.

If you do not specify a record length, the members are copied in sector mode format.

**SVATTR** This specifies that the library member attributes or indicators (MRT, PDATA, DATE, TIME, REF, SUB, and HIST) are saved when the member is copied in record mode format. MRT indicates that the library member is a multiple requester terminal procedure. PDATA indicates that the library procedure member passes parameters or program data when the procedure command processes. DATE indicates the date the member was created or last changed. TIME indicates the time the member was created or last changed. REF indicates the reference number of the member. SUB indicates the subtype of the member. HIST indicates that the library procedure member OCL statements are logged to the job log.

SVATTR is valid only if you also specify a record length. Specify a record length of at least 73 bytes if you specify SVATTR, or an error message is issued.

This parameter is not necessary when you copy members in sector mode format. The attributes are always saved in sector mode.

### Example 1

The following example saves a library procedure member named TEST on a diskette having a volume ID of VOL003. The procedure is in the library named MYLIB.

```
FROMLIBR TEST,PROC,,I1,,VOL003,MYLIB
```

### Example 2

The following example copies all library members to a permanent disk file. This example assumes that all payroll application source programs are in a library named MYLIB and begin with the characters PAY. The file's label is PAY, and the diskette's volume ID is VOL001.

```
FROMLIBR PAY,ALL,,,I1,999,VOL001,MYLIB
```

### Example 3

The following example copies a library source member named SAMPLE to a new disk file named FILE1. The source member is in the library named MYLIB, and the disk file is 4 blocks in size.

```
FROMLIBR SAMPLE,,FILE1,F1,,4,MYLIB
```

### Example 4

The following example saves a library procedure member named TEST on a tape with a volume ID of VOL001. The procedure is in the library named MYLIB and the tape reel is mounted on tape drive 1. After saving the procedure on tape, unload the tape.

```
FROMLIBR TEST,PROC,,T1,,VOL001,MYLIB,,,UNLOAD
```

### Example 5

The following example copies all source and procedure members in record mode format with a record length of 80, saving the member attributes:

```
FROMLIBR ALL,LIBRARY,FILE1,I1,999,VOL001,MYLIB,,80,SVATTR
```

## HELP Procedure

The HELP procedure allows you to:

- Display menus from which you can select options to do a desired task.
- Enter and run procedures. You can run the following procedures using HELP:
  - System/36 environment procedures (such as LISTLIBR or CATALOG).
  - Procedures that are part of the data file utility (DFU), screen design aid (SDA), and source entry utility (SEU).
  - Procedures that run the language licensed programs. This includes COBOL and RPG.
- Display information for the following System/36 environment functions:
  - OCL statements (HELP OCL)
  - Procedure control expressions (HELP PCE)

**Note:** On System/36, enter the name of the command and press F4 (Prompt) to see a prompt display that allows you to fill in the parameters. This is not supported in the System/36 environment *except* for the STATUS PRT command.

When you use the HELP procedure to specify parameters for a system procedure, and a user procedure with the same name exists in the current user library, the system procedure help display shows, but the user procedure runs. This results in the system procedure parameters being used by the user procedure. Unless the parameters match, the user procedure fails to run properly.

The HELP procedure runs the \$HELP utility program.

```
HELP [ procedure name [ ,parm1,parm2,... ]
      menu name
      MENUNAME
      OCL
      PCE ]
```

59020661-0

If you do not specify any parameters, the User Support display appears.

#### **procedure name**

This specifies the procedure for which you are prompted for parameters. The HELP procedure does not support the library parameter specified on the INCLUDE OCL statement. Refer to “INCLUDE OCL Statement” on page 5-49 for information on the library parameter for a procedure command.

#### **parm1,parm2,...**

This represents parameters for the procedure or control command, which help places on the prompt displays. Enter either one or more

## HELP

blanks, or a comma, between the procedure or control command name and the first parameter. For example, enter either of the following and the effect is the same:

```
HELP BLDLIBR MYLIB
```

or:

```
HELP BLDLIBR,MYLIB
```

You can also type:

```
BLDLIBR MYLIB
```

and press the F4 (Prompt) key.

All these cause the BLDLIBR procedure prompt display to show. The first parameter on the display is MYLIB.

### **menu name**

This specifies one of the system menus.

### **MENUNAME**

This returns to the AS/400 Main Menu display from any of the system menus.

### **OCL**

This specifies the display of a list of System/36 environment OCL statements. From that list, you can select a statement for which reference information displays.

**Note:** Entering OCL and pressing the F4 (Prompt) key results in an error. To request OCL you must enter HELP OCL.

### **PCE**

This specifies that a list of System/36 environment procedure control expressions displays. From that list, select the expressions for which reference information displays.

**Note:** Entering PCE and pressing the F4 (Prompt) key results in an error. To request PCE you must enter HELP PCE.

### **Example 1**

The User Support and Education display appears when you enter HELP without specifying a menu or procedure name. The User Support and Education display explains how to do such things as use help and commands, and how to handle problems. Your display appears as shown in the following example:

```

SUPPORT          User Support and Education          System: RCHAS1
Select one of the following:

    1. How to use help
    2. Search system help index
    3. How to use commands
    4. Question and answer
    5. AS/400 publications

    7. How to handle system problems
    8. Problem handling
    9. Online education

Selection or command
====>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F16=System main menu

```

## Example 2

The AS/400 Main Menu display appears when you enter HELP MAIN. This example shows how to use the menus to create a new library when you do not know the name of the System/36 environment procedure that creates a library. Your display appears as shown in the following example:

```

MAIN              AS/400 Main Menu          System: RCHAS1
Select one of the following:

    1. User tasks
    2. Office tasks
    3. General system tasks
    4. Files, libraries, and folders
    5. Programming
    6. Communications
    7. Define or change the system
    8. Problem handling
    9. Display a menu
    10. User support and education

    90. Sign off

Selection or command
====>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=User support
F23=Set initial menu

```

## HELP

Because you want to create a library, select option 4 to work with files, libraries, and folders. Type 4 and press the Enter key. The Files, Libraries, and Folders display appears as shown in the following example:

```
DATA                               Files, Libraries, and Folders          System: RCHAS1
Select one of the following:

    1. Files
    2. Libraries
    3. Folders
    4. Host system tasks for AS/400 PC Support

Selection or command
====>

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=User support
F16=System main menu

(C) COPYRIGHT IBM CORP. 1980, 1989.
```

Because you want to create a library, select option 2 to work with libraries. Type 2 and press the Enter key. The Libraries display appears as shown in the following example:

```
LIBRARY                               Libraries          System: RCHAS1
Select one of the following:

    1. Work with libraries
    2. Create a library
    3. Save a library
    4. Restore a library

    50. Save library members in System/36 format
    51. Restore library members from System/36 format
    52. System/36 library procedures

    70. Related commands

Selection or command
====> 52

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F13=User support
F16=System main menu

(C) COPYRIGHT IBM CORP. 1980, 1989.
```

Because you want to run a System/36 environment procedure to create a library, select option 52 to work with System/36 library procedures. Type 52 and press the Enter key. The System/36 Library Procedures display appears as in the following example:

```

S36LIBRARY                System/36 Library Procedures
                                System:  RCHAS1
Select one of the following:

    1. Create a library                BLDLIBR
    2. List names of libraries          CATALOG
    3. Rename member or change directory information  CHNGEMEM
    4. Remove a library from disk or diskette        DELETE
    5. Copy members from a library to a file         FROMLIBR
    6. Copy members from one library to another     LIBRLIBR
    7. List a library from diskette or tape         LISTFILE
    8. List the contents of a library               LISTLIBR
    9. Remove library members                   REMOVE
   10. Rename a library                         RENAME
   11. Restore a library to disk                  RESTLIBR
   12. Save library from disk to diskette or tape  SAVELIBR
   13. Change the session library                 SLIB
                                                More...

Selection or command
====>

F3=Exit  F4=Prompt  F5=Library menu  F9=Retrieve  F12=Cancel
F16=System main menu
(C) COPYRIGHT IBM CORP. 1980, 1989.

```

To create the library, type 1 and press the Enter key. The BLDLIBR Procedure display appears as in the following example:

```

                                BLDLIBR Procedure
                                Optional-*
                                Ignored-%

                Creates a new library with the option to copy members into it

Name of library to be created . . . . .
Size of library in blocks . . . . . 1-15000      %
Size of directory in sectors . . . . . 2-2500    * %
Preferred disk location . . . . . A1,A2,A3,A4,block number  * %
To copy a file into the library, enter file name . . . . . *

F3=Exit                                (C) COPYRIGHT IBM CORP. 1986, 1988.

```

### Example 3

The following example creates a library when you know that the BLDLIBR procedure creates libraries, but you do not know what parameters the BLDLIBR procedure requires. Enter the following to cause the BLDLIBR procedure prompt display to appear, and then fill in the parameters:

```
HELP BLDLIBR
```

You can also enter the following:

```
BLDLIBR
```

and press F4 (Prompt).

### Example 4

The example below creates a library named MYLIB when you know the following:

- The BLDLIBR procedure creates libraries.
- The first parameter of the BLDLIBR procedure is the name of the library.

But you do not know the other parameters.

Enter the following to cause the BLDLIBR prompt display to appear with the library name filled in. Then fill in the other parameters:

```
HELP BLDLIBR MYLIB
```

or:

```
HELP BLDLIBR,MYLIB
```

You can also enter:

```
BLDLIBR MYLIB
```

and press F4 (Prompt).

---

## HISTCOPY Procedure

Not supported. On System/36, the HISTCOPY procedure ran after the history file was automatically copied to a disk file named HISTCOPY. You could edit or change the HISTCOPY procedure to run your own job after the history file was copied.

On the AS/400 system, this similar function is not required.

---

## HISTCRT Procedure

The IBM System/34 HISTCRT procedure is not supported.

To do a similar function, use the Display Log (DSPLOG) and Display Job Log (DSPJOBLOG) commands. Select these commands from the Log Commands (CMDLOG) menu. You can also use the Work with Job (WRKJOB) command to display more detailed information about a job.

See the *CL Reference* book for more information on the DSPLOG, DSPJOBLOG, and WRKJOB commands. See the *System Operation* book for more information on history and job log information.

---

## HISTORY Procedure

Not supported. On System/36, the HISTORY procedure allowed you to:

- Display or print all or part of the history file or files created by the HISTCOPY procedure.
- Copy the history file to a disk file for later use.
- Erase the history file.

To do a similar function, use the Display Log (DSPLOG) and Display Job Log (DSPJOBLOG) commands. Select these commands from the Log Commands (CMDLOG) menu. You can also use the Work with Job (WRKJOB) command to display more detailed information about a job.

See the *CL Reference* book for more information on the DSPLOG, DSPJOBLOG, and WRKJOB commands. See the *System Operation* book for more information on history and job log information.

---

## ICFDEBUG Procedure

Not supported. On System/36, the ICFDEBUG procedure controlled the running of the Interactive Communications Feature (SSP-ICF) debug program.

To perform a similar function on the AS/400 system, use the Trace ICF (TRCICF) command. TRCICF provides the equivalent function of ICFDEBUG on a job basis as opposed to a system-wide basis. If you need to monitor several jobs, TRCICF must be entered several times, once in each job.

See the *System/36 Environment Programming* book for more information on debugging information for System/36 ICFDEBUG users. See the *CL Programming* book for more information on using the AS/400 program debugging functions. See the *ICF Programming* book for more information on communications error handling.

---

## ICVERIFY Procedure

Not supported. On System/36, the ICVERIFY procedure verified the installation of the Interactive Communications Feature (SSP-ICF) subsystems. It allowed you to check that the configuration parameters at both your system and at the remote system were correct, and verified that the communications line was connected and working.

There is no similar function available on the AS/400 system.

---

## IDDU Procedure

The IDDU procedure displays the interactive data definition utility (IDDU) system menu.

See the *IDDU Use* book for more information about the OS/400 IDDU.



59020648-0

The IDDU procedure has no parameters.

---

### IDDUDCT Procedure

The IDDUDCT procedure calls the interactive data definition utility (IDDU) to allow you to create, change, delete, or print a data dictionary.

See the *IDDU Use* book for more information about the OS/400 IDDU.



The IDDUDCT procedure has no parameters.

---

### IDDUDFN Procedure

The IDDUDFN procedure calls the interactive data definition utility (IDDU) to allow you to create, change, copy, delete, rename, print, or show a list of where a field, format, or file definition is used.

See the *IDDU Use* book for more information about the OS/400 IDDU.



The IDDUDFN procedure has no parameters.

---

### IDDUDISK Procedure

The IDDUDISK procedure calls the interactive data definition utility (IDDU) to allow you to create, enter, or update data in a disk file.

See the *IDDU Use* book for more information about the OS/400 IDDU.



The IDDUDISK procedure has no parameters.

---

### IDDULINK Procedure

The IDDULINK procedure links or unlinks a file on disk with a file definition in a data dictionary. Linking to the file definition allows a program or utility to access the file as it is defined by the file definition.

See the *IDDU Use* book for more information on the OS/400 IDDU.

To link a file on disk to a file definition:

```
IDDULINK LINK,file name,data dictionary name,file definition, [ mmddy
ddmmy
yyymmdd ]
```

59020138-0

To unlink a file on disk from a file definition:

```
IDDULINK UNLINK,file name,, [ mmddy
ddmmy
yyymmdd ]
```

59020479-1

To unlink all the file definitions in a data dictionary:

```
IDDULINK UNLINK,ALL,data dictionary name
```

59020480-0

**LINK** This specifies that the file on disk is associated with the file definition in the data dictionary.

**UNLINK** This specifies the removal of the association between the file on disk and the file definition.

**file name** This specifies the name of the database file on disk to link or unlink.

**ALL** This specifies the unlinking of all file definitions in the specified data dictionary from their associated database files on disk. You can specify ALL only if you specify UNLINK.

**data dictionary name**

This specifies the data dictionary that contains the file definitions to link or unlink. If the first parameter is UNLINK and the second parameter is a file name, this parameter is not required.

**file definition**

This specifies the name of the file definition to be linked to the database file. You can specify file definition only if you specify LINK.

**mmddy, ddmmy, or yyymmdd**

This is supported for System/36 compatibility only. The value is not used.

---

## IDDUPRT Procedure

The IDDUPRT procedure prints field, format, or file definitions in a data dictionary.

See the *IDDU Use* book for more information on the OS/400 IDDU.

```
IDDUPRT data dictionary name, [ SHORT
EXTENDED
ALL ], [ FILE
FORMAT
FIELD ], [ ALL
definition name ]
```

59020662-0

**data dictionary name**

This specifies the data dictionary that contains the definitions to print.

**SHORT** This specifies the printing of the short version of the definition. The short version always includes the text. The short version of a field definition includes the data type, field length, buffer length, column headings, and number of decimal places allowed in the field. A short format definition includes the number of fields, record length, and field list. A short file definition includes the number of record formats, type of file, and record format list. If you do not specify a parameter, **SHORT** is assumed.

**EXTENDED**

This specifies the printing of the extended version of the definition. The extended version always includes all the information supplied in a short definition plus a long comment and a list of where the definition is used. The extended version of a field definition includes additional attributes and numeric editing. An extended format definition includes record identification codes.

**ALL** This specifies the printing of the extended version of the definition. For a file definition, all extended versions of format definitions within the file are also printed. For a format definition, all extended versions of field definitions within the format are also printed.

**FILE** This specifies the printing of one or all file definitions within the specified data dictionary. If you do not specify a parameter, **FILE** is assumed.

**FORMAT** This specifies the printing of one or all format definitions within the specified data dictionary.

**FIELD** This specifies the printing of one or all field definitions within the specified data dictionary.

**ALL** This specifies the printing of all definitions within the specified data dictionary of the specified type. If you do not specify a parameter, **ALL** is assumed.

**definition name**

This specifies the name of a single definition of the specified type to print.

---

**IDDURBLD Procedure**

The **IDDURBLD** procedure is supported as a no-operation command. Only syntax checking of the procedure parameters is done.

On System/36, the **IDDURBLD** procedure allowed programs and utilities to use updated data definitions. Changes made to a definition were normally applied when the interactive data definition utility (**IDDU**) ended. If a program or utility was using the definition, however, you could not make changes until the program or utility ended. The **IDDURBLD** procedure was used after the program ended to apply the changes that **IDDU** could not apply when it ended.

On the AS/400 system, this function is not required.

See the *IDDU Use* book for more information about the OS/400 **IDDU**.

---

## IDDUXLAT Procedure

Not supported. On System/36, the IDDUXLAT procedure translated the RPG source specifications contained in RPG program source members or in Text Management System (TMS) data definitions into the interactive data definition (IDDU) definitions.

No similar function is available on the AS/400 system.

See the *IDDU Use* book for more information about the OS/400 IDDU.

---

## IGC Procedure

The IGC procedure allows you to set the System/36 environment double-byte character set (DBCS) job attribute. The support provided by this procedure is similar to the *IGC Session* prompt on the System/36 sign-on display. See the *System/36 Environment Programming* book for information on how to use the IGC procedure.

```
IGC  DBCS job attribute
```

59020649-0

### DBCS job attribute

**YES** specifies the job will be a System/36 environment DBCS job.

**NO** specifies the job will *not* be a System/36 environment DBCS job.

---

## INIT Procedure

The INIT procedure prepares one or more diskettes for use to save files and libraries. This preparation is called **initialization**. You can also rename or erase a diskette.

The INIT procedure does some or all the following functions:

- Writes identifying names on the diskette called the volume ID (also called the volume identifier or pack ID) and the owner ID
- Formats the control portion of the diskette (cylinder 0)
- Ensures that the diskette is usable
- Writes sector addresses on the diskette

The INIT procedure runs the \$INIT utility program.

```
INIT  [ volume id ] , [ owner id ] , [ RENAME ] , [ starting location ] ,
      [ program date ] , [ OWNERID ] , [ DELETE ] , [ S1 ]
      [ ] , [ ] , [ FORMAT ] , [ S2 ]
      [ ] , [ ] , [ FORMAT2 ] , [ S3 ]
      [ ] , [ ] , [ ] , [ M1.nn ]
      [ ] , [ ] , [ ] , [ M2.nn ]

      [ ending location ]
      [ starting location ]
      [ S1 ]
      [ S2 ]
      [ S3 ]
      [ M1.nn ]
      [ M2.nn ]
```

59020141-0

**volume id**

This specifies the following:

- The volume ID to assign the diskette, if you specify RENAME, FORMAT, or FORMAT2. The volume ID is from 1 through 6 characters (either alphabetic or numeric), and cannot contain special characters, such as \$, #, and @. The volume ID is placed in the volume ID field of the diskette volume label. If you do not specify a volume ID, the program date (job step date) is written in the volume ID field. The date is written in yymmdd (year-month-day) format.
- The volume ID of the diskette from which you delete files, if you specify DELETE. The volume ID you specify is checked against the existing volume ID of the diskette to ensure the insertion of the correct diskette.

**owner id** This specifies the owner ID of the diskette. The owner ID can be from 1 through 14 characters (either alphabetic or numeric) and cannot contain special characters, such as \$, #, and @. If you specify an owner ID and also specify RENAME, FORMAT, or FORMAT2, the owner ID is placed in the owner ID field of the diskette volume label. If you do not specify an owner ID, the word OWNERID is written in the owner ID field.

**RENAME** This specifies the renaming of the diskette (that is, the volume ID and owner ID fields in the diskette are assigned new values). It does not affect any files on the diskette. If you do not specify a parameter, RENAME is assumed.

**DELETE** This specifies the removal of any active files on the diskette, thereby making room on the diskette to hold more information.

**FORMAT** This specifies how the surface of the diskette is initialized. For a diskette 1 diskette, the diskette formats in the 128-byte per sector format. For a diskette 2D diskette, the diskette formats in the 256-byte per sector format.

If you specify FORMAT, the diskette can contain basic data exchange files. For more information about basic data exchange files, see "TRANSFER Procedure" on page 4-289.

**FORMAT2**

This specifies how the surface of the diskette is initialized. For a diskette 1 diskette, the diskette formats in the 512-byte per sector format. For a diskette 2D diskette, the diskette formats in the 1024-byte per sector format. Use this format as a general rule. FORMAT2 diskettes cannot contain basic data exchange files.

**starting location and ending location**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

Valid values for the starting and ending locations are as follows:

- S1, S2, or S3
- M1 or M2
- M1.nn or M2.nn (nn is a magazine slot number from 1 through 10)

---

## INITDIAG Procedure

Not supported. On System/36, the INITDIAG procedure initialized diagnostic (microcode) diskettes. These diskettes could then be used by the COPYDIAG procedure.

To do a similar function, use the Initialize Diskette (INZDKT) command.

See the *CL Reference* book for more information on the INZDKT command.

---

## INIT9332 Procedure

Not supported. On System/36, the INIT9332 procedure initialized and formatted the 9332 Disk Unit to allow the 9332 Disk Unit to transfer data with the system.

See the *9404 Models 1xx and Bxx-Fxx Installation and Upgrade* book, the *Backup and Recovery – Advanced* book, and *AS/400 Licensed Internal Code Diagnostic Aids – Volume 1* book for more information on preparing a disk unit for an AS/400 system.

---

## INQUIRY Procedure

The INQUIRY procedure allows you to create or change a DFU program to display a disk file using the data file utility (DFU). For more information about DFU, see the *ADTS/400: Data File Utility* book.

```
INQUIRY file name,dfu program name,[file source member name],,
      [
        D
        Z
        B
      ],[
        NN
        NY
        YN
        YY
        GO
      ],[dfu source member name],,
      [
        library name
        current library
      ],[display source member name],[name of file on disk]
```

S9020142-0

**file name** This specifies the file to display. The file name has from 1 through 8 alphanumeric characters.

### dfu program name

This specifies the DFU program used to process the file. If the program does not exist in the library, DFU starts the setup procedures to create the program. If the program exists in the library specified, DFU runs it. To change an existing DFU program, leave this name blank and specify the name of the program to be changed in the DFU source member name parameter. If both the DFU program name and the DFU source member name are not provided, DFU starts the setup procedures to create a temporary program.

See the “INQUIRY# Procedure” on page 4-123 for alternate methods of running a DFU program.

**file source member name**

This specifies the RPG II source member containing the file description (F-specifications) and record input specifications (I-specifications) that describe the file to process. This member can contain one or more sets of file description and input specifications, or an entire RPG II program. The file description and input specifications that correspond to the file make up the data description.

You receive a prompt for this parameter if you do not specify it. This parameter is required.

**D, Z, or B**

This indicates whether unkeyed selected zoned numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are D, Z, or B. If no parameter is specified, D is assumed.

**D or B** This specifies a data file with blank fill of unkeyed selected zoned fields.

**Z** This specifies a data file with zero fill of unkeyed selected zoned numeric fields.

**NN, NY, YN, YY, or GO**

This is supported for System/36 compatibility only. The value is not used.

**DFU source member name**

To change an existing DFU program, specify the name of the DFU program in this parameter and leave the DFU program parameter blank. If the DFU program is specified, then this parameter is ignored.

**library name**

This specifies the library that contains or will contain the DFU program. All library members associated with the DFU job are looked for, or stored, in this library. If you do not specify this parameter, the current library is assumed.

**display source member name**

This parameter is not supported. If a value is specified, an error message will be issued.

**name of file on disk**

This specifies the name in the disk VTOC of the file to display, if different from the name specified in the DFU program. If you specify this parameter, you can have several programs that refer to different files display the same file actually on disk. This is an optional parameter. If you specify the name of a file on disk and fail to specify a file that the DFU program displays, you receive a prompt for that parameter.

**Example**

The following example creates a program called DFIL1FMT which will display a disk file named FILE1. The program will be in the current library. The RPG II source member is RPGSRC and must already exist in the current library.

```
INQUIRY FILE1,DFIL1FMT,RPGSRC
```

## INQUIRY# Procedure

The INQUIRY# procedure allows you to display a disk file using the data file utility (DFU) by running the specified DFU program. Use RPG II file specifications to define the format of the records in the file. For more information about DFU, see the *ADTS/400: Data File Utility* book.

```
INQUIRY# file name,dfu program name,[file source member name],,
      [D
      Z
      B], [
      NN
      NY
      YN
      YY
      GO], [dfu source member name],,
      [
      library name
      current library], [display source member name], [name of file on disk]
```

S9020655-0

**file name** This specifies the file to display. The file name has from 1 through 8 alphameric characters.

### dfu program name

This specifies the DFU program used to display the file. This program must already exist in the specified library.

### file source member name

This is supported for System/36 compatibility only. The value is not used.

### D, Z, or B

This indicates whether unkeyed selected zoned numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are D, Z, or B. If no parameter is specified, D is assumed.

**D or B** This specifies a data file with blank fill of unkeyed selected zoned numeric fields.

**Z** This specifies a data file with zero fill of unkeyed selected zoned numeric fields.

### NN, NY, YN, YY, or GO

This is supported for System/36 compatibility only. The value is not used.

### DFU source member name

This is supported for System/36 compatibility only. The value is not used.

### library name

This specifies the library that contains or will contain the DFU program. All library members associated with the DFU job are looked for, or stored, in this library. If you do not specify this parameter, the current library is assumed.

### display source member name

This is supported for System/36 compatibility only. The value is not used.

**name of file on disk**

This specifies the name in the disk VTOC of the file to display, if different from the name specified in the DFU program. If you specify this parameter, you can have several programs that refer to different files display the same file actually on disk. This is an optional parameter. If you specify the name of a file on disk and fail to specify a file that the DFU program displays, you receive a prompt for that parameter.

**Example**

The following example displays a disk file named FILE1. The program is in the current library.

```
INQUIRY# FILE1,DFIL1FMT
```

---

**IPL Procedure**

Not supported. On System/36, the IPL procedure let you initiate an IPL of your system through program control. You could use this procedure to:

- Perform an IPL from disk without running through the IPL hardware diagnostics
- Load the system library from diskette again
- Load the system library from tape again
- Update the system microcode

To do similar function, use the Power Down System (PWRDWNSYS) command with the RESTART(\*YES) parameter specified.

**Note:** The PWRDWNSYS command does not check for active jobs before powering down the system.

See the *CL Reference* book for more information on the PWRDWNSYS command. See the *Backup and Recovery – Advanced* book for more information on how to reload your system libraries, including microcode.

---

**ITF Procedure**

The interactive terminal facility (ITF) procedure allows you to send and receive data through applications such as Telemail\*\* and the electronic message service offered by GTE Telenet for asynchronous terminals. Before you can start ITF, you must use the Vary Configuration (VRYCFG) command to vary on the line, controller, and device description. See the *Asynchronous Communications Programming* book and the *System/36 Environment Programming* book for more information about ITF.

```
ITF location name
```

59020530-0

**location name**

This specifies the name of the remote location with which you want to communicate.

### Example

The following example shows how to communicate with a remote location name of Chicago:

```
ITF CHICAGO
```

---

## IWLOAD Procedure

Not supported. On System/36, the IWLOAD procedure installed the PC Support/36 from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## IWPTLOAD Procedure

Not supported. On System/36, the IWPTLOAD procedure installed the PC Support/36 pass-through support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## IWPTSAVE Procedure

Not supported. On System/36, the IWPTSAVE procedure saved the PC Support/36 pass-through support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## IWSAVE Procedure

Not supported. On System/36, the IWSAVE procedure saved the PC Support/36 to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## JOBSTR Procedure

The JOBSTR procedure copies a diskette, tape, or tape cartridge file that contains one or more procedure members and source members to a specified library. You can also specify the name of a procedure to run after the members in the diskette, tape, or tape cartridge file are copied.

The file containing the procedure and/or source members must be readable by the JOBSTR procedure. The file can be either a **sector-mode** file or a **record-mode** file.

Sector-mode files and record-mode files are created by the FROMLIBR procedure. See "FROMLIBR Procedure" on page 4-103 for more information about creating these files. The JOBSTR procedure can also process record mode files created on other systems as long as those files follow the System/36 environment format.

You can store only source and procedure members as record-mode files. Records in a record-mode file can be from 40 to 120 characters long, but all records in a file must have the same length. Blanks are put at the end of lines that are shorter than the record length or records are truncated to the specified record length. The members in a record-mode file are preceded by a COPY statement and are fol-

lowed by a CEND statement. See “COPY and CEND Statements” on page A-47 for information about these statements.

JOBSTR runs the \$MAINT utility program and can process a JOBQ OCL statement.

```
JOBSTR file name, [ procedure name ], [ SAVE NOSAVE ], [ library name current library ],
[ Q jobq prty ], [ S1 S2 S3 M1.nn M2.nn ], [ AUTO NOAUTO ], [ I1 T1 T2 TC ], [ REWIND LEAVE UNLOAD ]
```

**file name** This specifies the name of the file containing one or more source or procedure members.

**procedure name**

This specifies the procedure to run after the members are copied. The procedure need not be one of the members to copy, but must be in the same library as the copied members.

**SAVE** This specifies that the procedure specified by **procedure name** remains in the library after it runs. If you do not specify a parameter, SAVE is assumed.

**NOSAVE** This specifies that after the procedure specified by **procedure name** runs, it is removed from the library.

**library name**

This specifies the library into which the members are copied. If you do not specify a library name, the current library is assumed.

**Q** This specifies that the procedure runs from the job queue, and has a job queue priority of 3. See the description of *jobq prty*, which follows, for a description of the job queue priority. If you do not specify a parameter but do specify a procedure name, the procedure runs as part of the job containing the JOBSTR procedure.

**jobq prty** This specifies the job queue priority for the job, or the job's order of processing from the job queue. The job queue priority can be any decimal number from 0 through 5. The system begins running jobs in order of decreasing job queue priority. For example, all jobs with a job queue priority of 5 run before any other jobs in the job queue. Jobs with the same job queue priority run in the order they were placed in the job queue. Jobs with a job queue priority of 1 are the last jobs run by the system. Job queue priority 0 is usually stopped. That is, any jobs placed on the job queue with a priority of 0 do not run until the system operator starts priority 0. If you do not specify a parameter but do specify a procedure name, the procedure runs as part of the job containing the JOBSTR procedure.

**S1, S2, or S3**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**M1.nn or M2.nn**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**AUTO** If specified for tape (unit is T1 or T2) when the tape reel on the original tape drive finishes, the system switches to the other tape drive to continue processing. If the tape reel on the other tape drive is not the same density as the original tape reel, an error message will be issued. If the other tape drive is not available, the system uses the original tape drive.

**NOAUTO** If specified for tape (unit is T1 or T2) when the tape reel on the original tape drive finishes, the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is tape and neither AUTO nor NOAUTO was specified, AUTO is assumed.

If TC is specified for the unit, the AUTO/NOAUTO parameter is ignored.

If the unit is diskette (I1), the AUTO/NOAUTO parameter is supported for compatibility only. If specified, the value will be syntax-checked, but will be ignored when the diskette is processed.

**I1** This specifies that the file to copy is on diskette.

**T1, T2, or TC**

This specifies that the file to copy is on tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

**REWIND** This specifies, for reel-to-reel tape, that the tape rewinds to the load point after processing completes.

For a tape cartridge, REWIND specifies that the tape is positioned to the beginning of the cartridge after processing completes.

**LEAVE** This specifies, for reel-to-reel tape or a tape cartridge, that tape is left where it was last processed. The next operation to the tape begins at that point.

**UNLOAD** This specifies, for reel-to-reel tape, that the tape rewinds and unloads after processing completes.

For a tape cartridge, UNLOAD specifies that the tape is positioned to the end of the cartridge after processing completes.

REWIND, LEAVE, and UNLOAD are valid only if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge processes or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, this process is lengthy. You can shorten the preparation time by specifying UNLOAD, because the tape is left at the end of the tape after the last operation completes.

## Examples

For the following examples, a diskette file labeled JOBS contains the following job stream:

```
// COPY NAME-PROC1,LIBRARY-P,HIST-NO
// LOAD PROGRAM
// RUN
// END
// CEND
// COPY NAME-PROC2,LIBRARY-P
// LOAD PROGRAM2
// FILE NAME-FILEABC
// RUN
// END
// CEND
// COPY NAME-MEMBER1,LIBRARY-S
THIS IS A SOURCE MEMBER
This is the second line of the source member
// CEND
```

### Example 1

The following example copies the procedure members (PROC1 and PROC2) and the source member (MEMBER1) into the library named MYLIB:

```
JOBSTR JOBS,,MYLIB
```

### Example 2

The following example copies the procedure members (PROC1 and PROC2) and the source member (MEMBER1) into a library called MYLIB and then run PROC1 (PROC1 should be saved):

```
JOBSTR JOBS,PROC1,,MYLIB
```

---

## KEYS Procedure

Not supported. On System/36, the KEYS procedure could perform the following functions:

- Display the current 3270-to-5250 keyboard map.
- Define a different set of keyboard mapping values for a 3270 keyboard.
- Change one or more of the keyboard mapping values for a 3270 keyboard.

To do a similar function, use the Change Keyboard Map (CHGKBDMAP) command, the Display Keyboard Map (DSPKBDMAP) command, and the Set Keyboard Map (SETKBDMAP) command. These commands can be selected from the Keyboard Commands (CMDKBD) menu.

See the *CL Reference* book for more information on the CHGKBDMAP, DSPKBDMAP, and SETKBDMAP commands.

---

## KEYSORT Procedure

The KEYSORT procedure is supported as a no-operation command. Only syntax checking of the procedure parameters is done. A check is made that the specified file does exist.

On System/36, the KEYSORT procedure sorted the keys for a specified indexed disk file. On the AS/400 system, this function is not required.

```
KEYSORT file name, [ mddyymmddyyymmdd ] , [ J ] , [ NOCHKDUP ]
                  [ ddmmyy ]           [ T ]   [ CHKDUP ]
                  [ yymmdd ]
```

S9020147-0

---

## LANLOAD Procedure

Not supported. On System/36, the LANLOAD procedure installed the LAN communications support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## LANSAVE Procedure

Not supported. On System/36, the LANSAVE procedure saved the LAN communications support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## LIBRLIBR Procedure

The LIBRLIBR procedure copies one or more library members from one library to another library. If you want to rename library members, see "CHNGEMEM Procedure" on page 4-40.

The LIBRLIBR procedure runs the \$MAINT utility program.

```
LIBRLIBR from library name, [ to library name ] , [ SOURCE ] ,
                          [ from library name ]   [ (S) ]
                                                  [ PROC ]
                                                  [ (P) ]
                                                  [ LOAD ]
                                                  [ (O) ]
                                                  [ SUBR ]
                                                  [ (R) ]
                                                  [ LIBRARY ]

[ member name ] , [ new member name ] , [ REPLACE ]
[ member name,ALL ]
[ ALL ]
```

S9020148-0

### from library name

This specifies the library from which to copy the one or more library members. If you do not specify a library name, you receive a prompt for it. The from library cannot be QSYS.

**to library name**

This specifies the library to which you copy one or more members. If you do not specify a library name, the from library name is assumed. The to library cannot be QSYS, QSSP, or QSYS38.

**Note:** If the from library name and the to library name parameters are the same, you must give the member being copied a new name.

**SOURCE or S**

This specifies the copying of source members only. If you do not specify a parameter, SOURCE is assumed. Source members are all the members in a source file named QS36SRC in the specified library.

**PROC or P**

This specifies the copying of procedure members only. Procedure members are all the members in a source file named QS36PRC in the specified library.

**LOAD or O**

This specifies the copying of load members only. Load members are all objects in a library with the following OS/400 object types:

**\*PGM** Programs  
**\*MSGF** Message files  
**\*FILE** Display files (attribute of \*DSPF)

**SUBR or R**

This specifies the copying of subroutine members only. Subroutine members are all objects in a library with the object type of \*PGM (programs).

**LIBRARY** This specifies the copying of all types of members (SOURCE, PROC, LOAD, and SUBR), including all objects with the following OS/400 object types:

**\*CHTFMT** Chart formats  
**\*CLS** Classes  
**\*CMD** Commands  
**\*DTAARA** Data areas  
**\*FCT** Forms control tables  
**\*FILE** Files (files with all attributes: PF, LF, PRTF, and so on)  
**\*GSS** Graphic symbol sets  
**\*JOB** Job descriptions  
**\*JOBQ** Job queues  
**\*MENU** Menus  
**\*MSGQ** Message queues  
**\*OUTQ** Output queues  
**\*QRYDFN** Query definitions  
**\*SBSD** Subsystem descriptions  
**\*SSND** Session descriptions  
**\*TBL** Tables

If a member type of LIBRARY is specified, the LIBRLIBR procedure copies AS/400 source files (such as QCLSRC and QRPGRSRC) entirely. When a LIBRLIBR command involving an AS/400 source file is entered, and the REPLACE keyword is specified or a zero option is taken to message SYS-2510, the target library source file with all its members is

deleted. The file in the source library is then copied into the target library.

Be aware that a LIBRLIBR of member type LIBRARY with the REPLACE option deletes all the existing members from each of the AS/400 target source files if a file of the same name exists in the source library.

**member name**

This specifies the library member to copy.

**member name,ALL**

This specifies the copying of all member names beginning with the characters specified by member name. Specify up to 7 characters for the member name.

**ALL**

This specifies the copying of all members. Do not specify ALL if the *from library* and the *to library* are the same.

**new member name**

This specifies the new member names of the copied members. Specify the new name if the *from library* and the *to library* are the same. If you specify **member name,ALL**, the new member name must contain the same number of characters as the member name. The remaining characters in each member name do not change.

If you do not specify a new member name, the names of the members do not change.

**REPLACE**

This specifies that, if the *to library* contains a library member with the same name as the member being copied, then the member being copied replaces the existing member.

If you do not specify REPLACE, the LIBRLIBR procedure copies only the new members.

**Example 1**

The following example copies the source member named TEST from a library named MYLIB into a library named YOURLIB:

```
LIBRLIBR MYLIB,YOURLIB,,TEST
```

**Example 2**

The following example copies all the library members named TEST from a library named MYLIB into a library named YOURLIB, and replaces any members named TEST in YOURLIB:

```
LIBRLIBR MYLIB,YOURLIB,LIBRARY,TEST,,REPLACE
```

**Example 3**

The following example copies all the procedures beginning with PAY from a library named TESTLIB to a library named PAYLIB. The copied members replace any existing procedure members in PAYLIB that begin with PAY.

```
LIBRLIBR TESTLIB,PAYLIB,P,PAY,ALL,,REPLACE
```

## LINES Procedure

The LINES procedure changes the number of lines per page, the horizontal print density (characters per inch), and the vertical print density (lines per inch) for printed output from a display station session. A job placed on the job queue uses the lines per page and the characters per inch values in effect when the job was placed on the queue.

You can also use the PRINT procedure to change these settings, see “PRINT Procedure” on page 4-178.

The LINES procedure processes a FORMS OCL statement.

```
LINES [ lines per page ], [ cpi value ], [ lpi value ]
```

59020149-0

### lines per page

This specifies the number of lines per page. Specify a maximum of 112 lines. If you do not specify a value, 66 is assumed. The lines per page specification remains in effect until either the session ends or you specify a new lines per page value:

- Another LINES procedure
- The PRINT procedure
- A FORMS OCL statement
- A PRINTER OCL statement (for that print file only)
- The SET procedure or the \$SETCF utility program

For most user-written programs and System/36 environment utility programs that use the system list device for output, the following is true:

- Printing starts on line 6 of the first page and on line 6 of all new pages.
- If the value specified for LINES is greater than 12, printing skips to a new page when six less than the number of lines specified is printed.

For example, if you specify LINES-66, printing starts on line 6 and skips to a new page after printing line 60. Thereafter, you get 55 lines of output per page (66 – 11 = 55, 5 blank lines at the top of a page and 6 blank lines at the bottom). If you specify LINES-13, two lines are printed per page. When you specify 12 or fewer lines per page, printing occurs on every line of the page.

For System/36 environment utility programs that use OS/400 print files for output, the following is true:

- Printing starts on line 1 of the first page and on line 1 of all new pages.
- If the value specified for LINES is greater than 12, printing skips to a new page when six less than the number of lines specified is printed.

For example, if you specify LINES-66, printing starts on line 1 and skips to a new page after printing line 60. Therefore, you get 60

lines of output per page ( $66 - 6 = 60$ , 6 blank lines at the bottom of the page). If you specify LINES-13, seven lines are printed per page. When you specify 12 or fewer lines per page, printing occurs on every line of the page.

**Note:** For a list of the System/36 environment utilities that use the system list device and the OS/400 print files, see the *System/36 Environment Programming* book.

For print operations from your programs, the System/36 environment indicates an overflow condition when you print six less than the number of lines specified.

**cpi value** This specifies the horizontal print density (that is, characters per inch) to use for printed output from the display station session. Specify the values 10 or 15. The printer in use must support the horizontal print density specified. If printer does not support the value specified, a message displays and you can continue or cancel the job. If you do not specify a characters per inch value, the horizontal print density does not change.

The characters per inch value remains in effect until either the session ends or you specify a new value by:

- Another LINES procedure
- The PRINT procedure
- A FORMS OCL statement
- A PRINTER OCL statement (for that print step only)

If you specify the switch option (0, zero) for the characters per inch value when the printer is configured, this parameter is ignored.

**lpi value** This specifies the vertical print density (lines per inch) to use for printed output from the display station session. If the output prints on a printer that does not support vertical print density, the lines per inch value is ignored.

The lines per inch value remains in effect until either the session ends or you specify a new lines per inch value using:

- Another LINES procedure
- The PRINT procedure
- A FORMS OCL statement
- A PRINTER OCL statement (for that print step only)

If you do not specify a lines per inch value and the value was not previously set during the session, the system uses the value set during work station configuration.

If you specify the switch option (0, zero) for the lines per inch value during printer configuration, this parameter is ignored.

### Example 1

The following example changes the number of lines per page to 33:

```
LINES 33
```

## Example 2

The following example changes the lines printed per page to 88, the horizontal print density to 15 characters per inch, and the vertical print density to 8 lines per inch. Use 11-inch high paper (11 x 8 = 88 lines per page).

LINES 88,15,8

---

## LIST Procedure

The LIST procedure allows you to print and also sort a disk file using the LIST component of the data file utility (DFU). You use file specifications to define the format of the records in the file.

```
LIST file name,dfu program name,[file source member name],
    [NOSORT], [D], [NN], [dfu source member name],
    [SORT], [Z], [NY],
    [B], [Y],
    [YY],
    [GO]
    [master file name], [library name], [name of file on disk]
    [current library]
```

59020150-0

**file name** This specifies the file to print. The file name can be from 1 through 8 alphameric characters.

### dfu program name

This specifies the DFU program used to process the file. If the program does not exist in the library, DFU starts the setup procedures to create the program. If the program exists in the library specified, DFU runs it. To change an existing DFU program, leave this name blank and specify the name of the program to be changed in the DFU source member name parameter. If both the DFU program name and the DFU source member name are not provided, DFU starts the setup procedures to create a temporary program.

See also the "LIST# Procedure" on page 4-135 for alternate methods of running a DFU program.

### file source member name

This specifies the RPG II source member containing file description (F-specifications) and record input specifications (I-specifications) that describe the file to process. This member can contain one or more sets of file description and input specifications, or an entire RPG II program. The file description and input specifications that correspond to the file are the data description.

This parameter is required.

**NOSORT** This specifies that the file is not sorted before it prints. If you do not specify a parameter, NOSORT is assumed.

**SORT** This specifies that the file is sorted before it prints.

**D, Z, or B**

This indicates whether unkeyed selected zoned numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are D, Z, or B. If no parameter is specified, D is assumed.

**D or B** This specifies a data file with blank fill of unkeyed selected zoned numeric fields.

**Z** This specifies a data file with zero fill of unkeyed selected zoned numeric fields.

**NN, NY, YN, YY, or GO**

This is supported for System/36 compatibility only. The value is not used.

**DFU source member name**

To change an existing DFU program, specify the name of the DFU program in this parameter and leave the DFU program parameter blank. If the DFU program is specified, then this parameter is ignored.

**master file name**

This specifies the indexed file that contains information related to the file.

**library name**

This specifies the library that contains or will contain the DFU program. All library members associated with the DFU list program are looked for, or stored, in this library. If you do not specify this parameter, the current library is assumed.

**name of file on disk**

This specifies the name in the disk VTOC of the file to print, if it is different from the name specified in the DFU program. If you specify this parameter, you can have several programs that refer to different logical files print the same file actually on disk. This is an optional parameter. If you specify the name of a file on disk and fail to specify a file to print using the DFU program, you are prompted for that parameter.

**Example**

The following example creates a program called DFIL1FMT to print a disk file named FILE1. The program will be in the current library and the file source member RPGSRC is also in the current library. The output will not be sorted.

```
LIST FILE1,DFIL1FMT,RPGSRC,NOSORT
```

---

## LIST# Procedure

The LIST# procedure allows you to print and also sort a disk file using the LIST component of the data file utility (DFU).

```

LIST#   file name,dfu program name,[file source member name],

        [NOSORT], [D], [NN], [dfu source member name],
        [SORT],  [Z], [NY],
        [B],    [Y], [Y],
               [GO]

        [master file name],[library name], [name of file on disk]
                           [current library]

```

59020656-0

**file name** This specifies the file to print. The file name can be from 1 through 8 alphameric characters.

**dfu program name**

This specifies the DFU program to use to print the file. This program must already exist in the specified library.

**file source member name**

This is supported for System/36 compatibility only. The value is not used.

**NOSORT** This specifies that the file is not sorted before it prints. If you do not specify a parameter, NOSORT is assumed.

**SORT** This specifies that the file is sorted before it prints.

**D, Z, or B**

This indicates whether unkeyed selected zoned numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are D, Z, or B. If no parameter is specified, D is assumed.

**D or B** This specifies a data file with blank fill of unkeyed selected zoned numeric fields.

**Z** This specifies a data file with zero fill of unkeyed selected zoned numeric fields.

**NN, NY, YN, YY, or GO**

This is supported for System/36 compatibility only. The value is not used.

**DFU source member name**

This is supported for System/36 compatibility only. The value is not used.

**master file name**

This specifies the indexed file that contains information related to the file.

**library name**

This specifies the library that contains or will contain the DFU program. All library members associated with the DFU list program are looked for, or stored, in this library. If you do not specify this parameter, the current library is assumed.

**name of file on disk**

This specifies the name in the disk VTOC of the file to print, if it is different from the name specified in the DFU program. If you specify this parameter, you can have several programs that refer to different logical

files print the same file actually on disk. This is an optional parameter. If you specify the name of a file on disk and fail to specify a file to print using the DFU program, you are prompted for that parameter.

### Example

The following example prints a disk file named FILE1. The program is in the current library.

```
LIST# FILE1,DFIL1FMT
```

---

## LISTDATA Procedure

The LISTDATA procedure lists, on the system list device assigned to the display station, all or part of a disk, tape, tape cartridge, or diskette file created using the \$COPY utility. During the list operation, the LISTDATA procedure can omit or include specific records. If you try to print with double-byte character set (DBCS) headings and data at a non-double-byte character set printer, blanks print in place of the DBCS characters.

In order to list the contents of a diskette, tape, or tape cartridge file created by the \$COPY utility, the LISTDATA procedure must temporarily restore the entire file to a temporary library on disk. A message is issued informing you of this whenever you attempt to list such a file.

When you use the LISTDATA procedure to change the contents of the file that you are displaying on your display station, a temporary intermediate file and library are created.

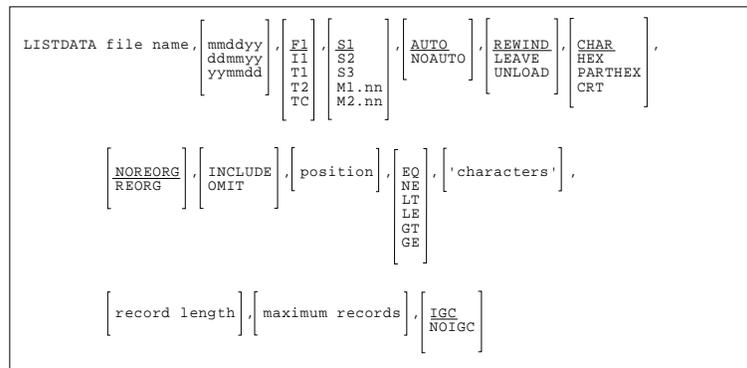
You can use the LISTDATA procedure to process diskette, tape, or tape cartridge files created using the \$COPY on the System/32, System/34, or System/36.

For information about assigning the system list utility device, see "SYSLIST Procedure" on page 4-253. When the system list device is the display station, (CRT), the information displays as if you had entered the CRT parameter.

The LISTDATA procedure can process a disk file that is being used by another job on the system, if that job's FILE statement specifies DISP-SHRRR or DISP-SHRRM. See "FILE OCL Statement (for Disk Files)" on page 5-25 for more information about the DISP parameter.

You can use the LISTDATA procedure to display or print the contents of a disk file located on a remote system.

The LISTDATA procedure runs the \$COPY utility program.



59020151-1

**file name** This specifies the file to be listed. The file can be of any file organization.

### **mmddyy, ddmmyy, or yymmdd**

This specifies the creation date of the file to be listed. The date, if specified, must be in the session date format. Use the STATUS SESSION command to determine the session date format. If you do not specify a date and more than one file exists with the same file name, the most recent file created is listed. If you specify a date for a diskette, tape or tape cartridge file, you must specify the date that the file was originally created on disk.

**F1** This specifies that the file to be listed is on disk. If you do not enter a parameter, F1 is assumed.

**I1** This specifies that the file to be listed is on diskette.

### **T1, T2, or TC**

This specifies that the file to be listed is on tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

### **S1, S2, S3, M1.nn, or M2.nn**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**AUTO** If specified for tape (unit is T1 or T2) when the tape reel on the original tape drive finishes, the system switches to the other tape drive to continue processing. If the tape reel on the other tape drive is not the same density as the original tape reel, an error message will be issued. If the other tape drive is not available, the system uses the original tape drive.

**NOAUTO** If specified for tape (unit is T1, T2, or TC) when the tape reel on the original tape drive finishes, the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is specified for tape and neither AUTO nor NOAUTO is specified, AUTO is assumed.

If TC is specified for the unit, the AUTO/NOAUTO parameters are ignored.

If the unit is diskette (I1), the AUTO/NOAUTO parameter is supported for compatibility only. If specified, the value will be syntax-checked, but will be ignored when the diskette is processed.

**REWIND** If specified for a reel-to-reel tape, the tape is rewound to the load point after processing completes.

If specified for a tape cartridge, the tape is positioned to the beginning of the cartridge after processing completes.

**LEAVE** If specified for a reel-to-reel tape, or a tape cartridge, the tape is left where it was last processed. The next operation to the tape will begin at that point.

**UNLOAD** If specified for a reel-to-reel tape drive, the tape is rewound and unloaded after processing completes.

If specified for a tape cartridge, the tape is positioned to the end of the cartridge after processing completes.

REWIND, LEAVE, or UNLOAD are only valid if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge is processed or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, the preparation is a lengthy process. The preparation time can be shortened if UNLOAD is specified, because the tape is left at the end of the tape (not rewound) after the last operation is completed.

#### **CHAR, PARTHEX, HEX, or CRT**

This specifies that the records are to be listed on the current system list device. If the current system list device is a printer, these parameters also specify how the records are to be printed. If the current system list device is the display station (CRT), the records are displayed as if you had specified the CRT parameter. The CHAR, PARTHEX, and HEX specifications are ignored because you can use the CRT specification to select these functions using command keys. Use the STATUS SESSION command to determine your current system list device. Use the SYSLIST procedure to change your system list device.

**CHAR** This specifies that if the current system list device is a printer, only the printable characters in each record should be printed. If you do not specify a parameter, CHAR is assumed. Any unprintable characters are printed as blanks. For example, the following record contains an unprintable character in the third position:

```
12 4567
```

#### **PARTHEX or HEX**

This specifies that if the current system list device is a printer, the printable characters and the hexadecimal representations of all characters should be printed. For example, the following record contains an unprintable character in the third position. That character's value is hex FA:

```
12 4567
FFFFFFF
12A4567
```

**CRT** This specifies that (no matter what the current system list device is) the records should be displayed at the display station. You cannot specify CRT for a job placed on the job queue.

You can display the records in both character format and hexadecimal format.

**REORG** This specifies that if the file to be listed is an indexed file, the records are listed sequentially by key. (Deleted records are never listed, regardless of whether the REORG or the NOREORG parameter is specified.)

**NOREORG**

This specifies that the records in the file are to be listed in the position where they occur in the file. (Deleted records are never listed regardless of whether the REORG or the NOREORG parameter is specified.)

**INCLUDE or OMIT**

This specifies whether specific records in the file are to be included in or are to be omitted from the listing. The INCLUDE and OMIT parameters work with the position, EQ, NE, LT, GT, LE, GE, and 'characters' parameters. If you specify only 'characters' or position, INCLUDE and EQ are assumed.

**position** This specifies, for each record, the first character to compare with the comparison characters. The position can be any number from 1 through 4096. If you do not specify a position, and INCLUDE and EQ have been specified (or can be assumed), every position in the record is compared with the comparison characters until the condition is met.

**EQ** This specifies that if the characters in the record indicated by position are the same as the comparison characters, the record is to be included in or omitted from the listing.

If you specify only characters or position, INCLUDE and EQ are assumed.

**NE** This specifies that if the characters in the record indicated by position are not the same as the comparison characters, the record is to be included in or omitted from the listing.

**LT** This specifies that if the characters in the record indicated by position are less than the comparison characters, the record is to be included in or omitted from the listing.

**LE** This specifies that if the characters in the record indicated by position are less than or the same as the comparison characters, the record is to be included in or omitted from the listing.

**GT** This specifies that if the characters in the record indicated by position are greater than the comparison characters, the record is to be included in or omitted from the listing.

**GE** This specifies that if the characters in the record indicated by position are greater than or the same as the comparison characters, the record is to be included in or omitted from the listing.

**'characters'**

This specifies the comparison characters. You can specify up to 30 characters and you should enclose them in apostrophes ('). You can

include blanks and commas (,), but you cannot include apostrophes as data.

### **record length**

This specifies the length of the records to be listed, and can be any number from 1 through 4096. If you do not enter this parameter, the entire record will be listed. You can use this parameter to list only a portion of each record in the file. If you are listing the contents of an alternative index file, you cannot request a length greater than the record length of the physical file upon which the alternative index file is based.

If the record length of the file to be listed is less than the specified record length, the additional record positions in the listing are filled with blanks. If the record length of the file to be listed is greater than the specified record length, the extra positions in the file are truncated when the file is listed.

### **maximum records**

This specifies the total number of records to be listed when you do not want to list the entire file. Enter any number greater than 0. This parameter is ignored when the entries are to be displayed at a display station.

**IGC** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**NOIGC** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**Note:** When printing the contents of a file on a double-byte character set (DBCS) system, the system determines whether the file may contain DBCS data. DBCS data cannot be displayed at a display station using the LISTDATA procedure.

### **Example 1**

The following example prints only the printable characters in all the records of the disk file named FILE1:

```
LISTDATA FILE1,,F1
```

### **Example 2**

The following example displays all the records in the disk file named FILE2, regardless of the current system list device:

```
LISTDATA FILE2,,,,,CRT
```

### **Example 3**

The following example prints all the records in the diskette file named FILE1. Both the printable characters in the records and the hexadecimal representations of all the characters in the records are printed.

```
LISTDATA FILE1,,I1,S1,NOAUTO,,HEX
```

**Example 4**

The following example prints only the records in the disk file named FILE3 that have the character X in position 5 of the record. Only the printable characters in the records are printed.

```
LISTDATA FILE3,,,,,,INCLUDE,5,EQ,'X'
```

**Example 5**

The following example displays a file named FILEA from the reel of the tape mounted on tape drive T1. When the display finishes, the tape reel rewinds to the beginning of the tape.

```
LISTDATA FILE4,,T1,,REWIND,CRT
```

**Example 6**

The following example displays the contents of a tape cartridge file named DATA1 on the system list device:

```
LISTDATA DATA1,,TC
```

---

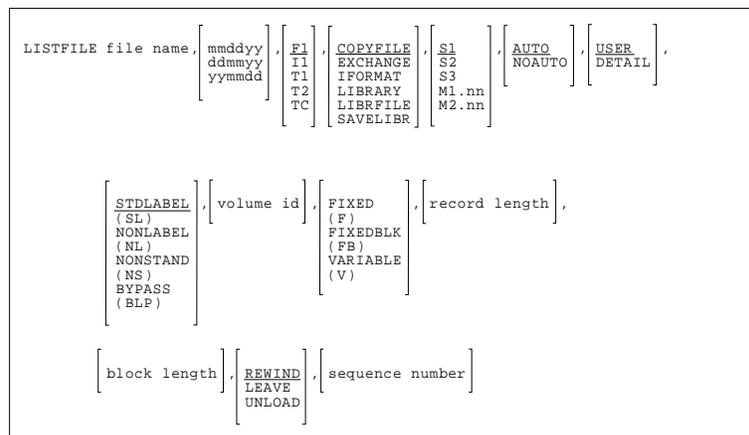
**LISTFILE Procedure**

The LISTFILE procedure lists the contents of a specified file or library on the system list device assigned to the requester display station. For information about assigning the system list device, see the “SYSLIST Procedure” on page 4-253.

In order to list the contents of a diskette, tape, or tape cartridge file created by the \$COPY utility, the LISTFILE procedure must restore the entire file to a temporary library on disk. Whenever you attempt to list such a file, a message will be issued to inform you that the file will be restored.

The LISTFILE procedure will process \$COPY diskette, tape, or tape cartridge files created on System/32, System/34, or System/36. If you try to print DBCS headings and data at a non-DBCS printer, blanks are printed in place of the double-byte characters.

The LISTFILE procedure runs the \$BICR, \$COPY, \$MAINT, or \$TCOPY utility program, depending on the parameters you enter.



**file name** This specifies the file to be listed.

**mmddy, ddmmy, or yymmdd**

This specifies the creation date of the file to be listed. For a disk file, the specified date must be in the same format as the session date. If you do not specify the creation date for a disk file, and more than one file exists with the specified file name, the file with the most recent creation date is assumed.

For a diskette, tape, or tape cartridge file other than a COPYFILE, put the specified date in the same format as the creation date of the diskette, tape, or tape cartridge file. If you specify a date for a COPYFILE located on diskette, tape or tape cartridge, you must specify the date that the file was originally created on disk in session date format. If you do not specify the creation date for a diskette, tape, or tape cartridge file, and more than one file exists with the specified file name, the first file in the diskette VTOC or on the tape is processed.

**F1** This specifies listing of a disk file. If you do not specify a parameter, and you specify LIBRARY, LIBRFILE, or COPYFILE, then F1 is assumed.

**I1** This specifies listing of a diskette file. If you do not specify a parameter, and you specify EXCHANGE, IFORMAT, or SAVELIBR, then I1 is assumed.

**T1, T2, or TC**

This specifies listing of a tape file. T1 indicates mounting of the tape on tape drive 1. T2 indicates mounting of the tape on tape drive 2. TC indicates that the tape is a tape cartridge.

**COPYFILE**

This specifies listing of a data file that is on disk, diskette, tape, or tape cartridge. The diskette, tape, or tape cartridge file must have been created by the \$COPY utility (the SAVE procedure). If you do not specify a parameter, COPYFILE is assumed.

**EXCHANGE**

This specifies listing of a basic data exchange diskette file or exchange tape file. See the "TRANSFER Procedure" on page 4-289 for information about basic data exchange files. If you specify EXCHANGE, F1 and TC are not allowed.

**IFORMAT**

This specifies listing of an I-exchange diskette file. See the "TRANSFER Procedure" on page 4-289 for information about I-exchange files. If you specify IFORMAT, then F1, T1, T2, and TC are not allowed.

**LIBRARY** This specifies listing of a library that is on disk. The file name specifies the name of the library. The listing is in the same format as a listing produced by the following LISTLIBR procedure:

```
LISTLIBR DIR,LIBRARY,file name
```

If you specify LIBRARY, then I1, T1, T2, and TC are not allowed.

**LIBRFILE** This specifies that the disk, diskette, tape or tape cartridge file to be listed was created by the FROMLIBR procedure. The listing that is generated will be similar to the listings generated by the OS/400 DSPDKT, DSPTAP or DSPSAVF commands.

## **SAVELIBR**

This specifies that the diskette, tape, or tape cartridge file to be listed was created by the SAVELIBR procedure. The listing that is generated will be similar to the listings generated by the OS/400 DSPDKT or DSPTAP commands.

## **S1, S2, S3, M1.nn, or M2.nn**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

## **AUTO**

If specified for tape (unit is T1 or T2) when the tape reel on the original tape drive finishes, the system switches to the other tape drive to continue processing. If the tape reel on the other tape drive is not the same density as the original tape reel, an error message will be issued. If the other tape drive is not available, the system uses the original tape drive.

## **NOAUTO**

If specified for tape (unit is T1, T2, or TC) when the tape reel on the original tape drive finishes, the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is tape and neither AUTO nor NOAUTO was specified, AUTO is assumed.

If TC is specified for the unit, the AUTO/NOAUTO parameter is ignored.

If the unit is diskette (11), the AUTO/NOAUTO parameter is supported for compatibility only. If specified, the value will be syntax-checked, but will be ignored when the diskette is processed.

## **USER or DETAIL**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

## **STDLABEL or SL**

This specifies that the tape or tape cartridge being processed has standard tape labels and that all the information needed to process the file can be found in the tape file label. If this parameter is not specified, STDLABEL is assumed. If you specify TC for parameter 3, you must either leave this parameter blank or specify STDLABEL (SL).

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

## **NONLABEL or NL**

This specifies that the tape being processed does not have any labels and that you must supply all the information needed to process the file. NONLABEL (NL) is invalid if you specify TC for parameter 3.

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

## **NONSTAND or NS**

This specifies that the tape being processed has nonstandard labels and that you must supply all the information needed to process the tape. NONSTAND (NS) is invalid if you specified TC for parameter 3.

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

**BYPASS or BLP**

This specifies that the tape being processed has standard labels and that label processing is bypassed. You must supply all the information needed to process the file. BYPASS (BLP) is invalid if you specified TC parameter 3.

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

**volume id**

This specifies the volume identification of the tape or tape cartridge being processed and can be 1 to 6 alphanumeric characters. The system uses this parameter to check that the correct tape volume is mounted. If the correct tape is not mounted, an error message is issued. You can continue to process the file, retry after mounting the correct tape, or cancel. This parameter is not allowed if you specify NONLABEL, NONSTAND, or BYPASS. If you are listing a multivolume tape, the system only checks the volume ID of the first.

**FIXED or F**

This specifies the record format of the tape file being processed as fixed length, unblocked records. If you do not specify the record format and you do specify EXCHANGE, FIXED is assumed. FIXED (F) is invalid if you specify TC.

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

**FIXEDBLK or FB**

This specifies the record format of the tape or tape cartridge file being processed as fixed length, blocked records. If you do not specify the record format, and the file is a COPYFILE, LIBRFILE, or SAVELIBR file, FIXEDBLK is assumed. If you specify TC, you must either leave this parameter blank or specify FB.

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

**VARIABLE or V**

This specifies the record format of the tape file being processed as variable length, unblocked records. VARIABLE (V) is invalid if you specified TC.

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

**Note:** If the type of processing is STDLABEL and the record format specified (or defaulted to) does not match the record format in the tape file label, an error message is issued.

**record length**

This specifies the number of bytes in a logical tape or tape cartridge record. For variable length records this is the maximum length. You can specify from 18 to 4096 bytes. The record length parameter is required if you specify NONLABEL, NONSTAND, or BYPASS. If you specify STDLABEL but do not specify the record length, the record length is taken from the tape or tape cartridge file label. If you specify both the record length and STDLABEL, the record length you specify must be the

same as the record length in the tape or tape cartridge file label. If you specify the record length when listing a COPYFILE file, it must be 256.

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

### **block length**

This specifies the number of bytes in a physical block of data of the tape or tape cartridge file. You can specify from 18 to 32 767 bytes. This parameter is required if you specify FIXEDBLK and NONLABEL, NONSTAND, or BYPASS. If you specify FIXEDBLK and STDLABEL, and do not specify the block length, the block length is taken from the tape or tape cartridge file label. If you specified the block length and both FIXEDBLK and STDLABEL, the block length must be the same as the block length in the tape file label. This parameter is ignored if you specify FIXED or VARIABLE. If you specify this parameter when you list a COPYFILE it must be 24 576.

If you use the LISTFILE procedure to list information about a library, this parameter is ignored.

**REWIND** If specified for a reel-to-reel tape, the tape is rewound to the load point after processing is complete.

If specified for a tape cartridge, the tape is rewound to the beginning of the cartridge after processing is complete.

**LEAVE** If specified for a reel-to-reel tape, or a tape cartridge, the tape is left where it was last processed. The next step within the same procedure which accesses the tape drive will start at this same position if you specify the same tape unit.

**UNLOAD** If specified for a reel-to-reel tape drive, the tape is rewound and unloaded after processing is complete.

If specified for a tape cartridge, the tape is positioned to the end of the cartridge after processing is complete.

REWIND, LEAVE, or UNLOAD are only valid if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge is processed or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, the preparation is a lengthy process. The preparation time can be shortened if UNLOAD is specified, because the tape is left at the end of the tape (not rewound) after the last operation is completed.

### **sequence number**

This specifies which file to process by its position on the tape or tape cartridge. Specify a number from 1 through 9999. If you omit this parameter, and specify NONLABEL, NONSTAND, or BYPASS, a sequence number of 1 is assumed. If you omit this parameter, and specify STDLABEL, the file is found by its file label. If you specify a sequence number and STDLABEL, the system locates this file by sequence number first, and then checks the file label. If it is not the correct file, the system locates the file by its file label.

**Example 1**

The following example lists the contents of a disk file named PAY:

```
LISTFILE PAY,,F1
```

**Example 2**

The following example lists the members of a library named MYLIB that is saved on diskette:

```
LISTFILE MYLIB,,I1,SAVELIBR
```

**Example 3**

The following example lists a diskette file named FILEA that contains several library members:

```
LISTFILE MYLIB,,I1,LIBRFILE
```

**Example 4**

The following example lists the contents of an exchange file named EXFILE, which is the third file on a nonlabeled tape of variable length, unblocked records. This example specifies that the tape should not rewind when processing is completed.

```
LISTFILE EXFILE,,T1,EXCHANGE,,,,NL,,V,132,1320,LEAVE,3
```

**Example 5**

The following example lists a SAVELIBR file named MYLIB from tape cartridge. The file was saved using the SAVELIBR procedure.

```
LISTFILE MYLIB,,TC,SAVELIBR
```

---

## LISTLIBR Procedure

The LISTLIBR procedure lists the contents of a library. You can list the following:

- The members contained in the library (the directory entries)
- The contents of the individual library members
- The directory information for members by name, partial name, or subtype
- The number of entries in a library

You can direct the listing to the system list device or to a disk file. If you specify an output file name, output is directed to that file. Otherwise, output is directed to the system list device. For information about assigning the system list device, see the "PRINT Procedure" on page 4-178 or the "SYSLIST Procedure" on page 4-253.

The LISTLIBR procedure runs the \$MAINT utility program.

To list the names of each member and other information (the directory entries):

```
LISTLIBR DIR, [SOURCE], [library name], [USER], [subtype],,
              (S)  [current library] [DETAIL]
              PROC
              (P)
              LOAD
              (O)
              SUBR
              (R)
              LIBRARY

              [output file name]
```

S9020153-1

To list the contents of one or more library members and their directory entries:

```
LISTLIBR {member name
         {member name,ALL
         ALL} , {SOURCE
                (S)
                PROC
                (P)
                LOAD
                (O)
                SUBR
                (R)
                LIBRARY} , {library name
                             current library} , {USER
                                                  DETAIL} , {NOPAGE
                                                             PAGE} ,

         {subtype} , {MEMBERS}
```

59020154-0

To list the directory information by name, partial name, or subtype:

```
LISTLIBR {member name
         {member name,ALL
         ALL} , {SOURCE
                (S)
                PROC
                (P)
                LOAD
                (O)
                SUBR
                (R)
                LIBRARY} , {library name
                             current library} , {USER
                                                  DETAIL} , {NOPAGE
                                                             PAGE} ,

         {subtype} , {DIRINFO} , {output file name}
```

59020155-1

To list only the status information about a library:

```
LISTLIBR DIR,SYSTEM, {library name
                      current library} ,,,,, {output file name}
```

59020156-1

**DIR** This specifies listing of the directory status information (if the second parameter is LIBRARY or SYSTEM) and the directory entries (if the second parameter is not SYSTEM).

**member name**

This specifies the library member to list.

**member name,ALL**

This specifies listing of the contents of the library members whose names begin with the specified characters (member name). Specify up to 7 characters for member name.

**ALL** This specifies listing of all members of the library.

**SOURCE or S**

This specifies listing of only library source members or directory entries. If you do not specify a parameter, SOURCE is assumed. Source members are all the members in a source file named QS36SRC in the specified library.

**PROC or P**

This specifies listing of only the library procedure members or directory entries. Procedure members are all the members in a source file named QS36PRC in the specified library.

**LOAD or O**

This specifies listing of only library load members or directory entries. Load members are all objects in a library with the following OS/400 object types:

<b>*PGM</b>	Programs
<b>*MSGF</b>	Message files
<b>*FILE</b>	Display files (attribute of *DSPF)

**SUBR or R**

This specifies listing of only the library subroutine members or directory entries. Subroutine members are all objects in a library with the OS/400 object type of \*PGM (programs).

**LIBRARY** This specifies listing of all library member types (SOURCE, PROC, LOAD, and SUBR), including all objects with the following OS/400 object types:

<b>*CHTFMT</b>	Chart formats
<b>*CLS</b>	Classes
<b>*CMD</b>	Commands
<b>*DTAARA</b>	Data areas
<b>*FCT</b>	Files (files with all attributes: PF, LF, PRTF, and so on)
<b>*GSS</b>	Graphic symbol sets
<b>*JOB</b>	Job descriptions
<b>*JOBQ</b>	Job queues
<b>*MENU</b>	Menus
<b>*MSGQ</b>	Message queues
<b>*OUTQ</b>	Output queues
<b>*QRYDFN</b>	Query definitions
<b>*SBSD</b>	Subsystem descriptions
<b>*SSND</b>	Session descriptions
<b>*TBL</b>	Tables

**SYSTEM** This specifies listing of only the number of entries in the library. No library member names are listed. SYSTEM is valid only with DIR.

**library name**

This specifies the library containing the members to list. If you do not specify a library name, the current library is assumed.

**USER or DETAIL**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**output file name**

This specifies the name of the disk file to which the output of the LISTLIBR procedure is written. The file name has from 1 to 8 alphanumeric characters and must not contain blanks, commas, or quotation marks. ALL is not a valid file name. If you do not specify an output file name, output writes to the SYSLIST device. The specified file is created as a sequential file on the local system. The record length is 132 bytes.

**Note:** A file or library with the specified name and the current date must not already exist on the disk.

**NOPAGE or PAGE**

This specifies, for source and procedure members only, if each member lists on a separate page. If you do not specify a parameter, NOPAGE is assumed. Load and subroutine members always list on separate pages.

**NOPAGE** This specifies that as many members should be listed per page as can fit. Use this option to save printer paper. See Figure 4-4 on page 4-154 for a sample of this listing.

**PAGE** This specifies the listing of each member on a separate page.

**subtype** This specifies the subtype of the members listed. If not specified, the subtype is not used as a qualifier when selecting members. The following table shows valid subtypes.

Subtype to Enter	Subtype Displayed	Meaning
ARP	RPT or RPT36	The RPG auto report member
ARS	ARS36	The automatic response member
ASM	ASM36	The assembler member
BAP	BASP or BASP36	The BASIC procedure (source member)
BAS	BAS or BAS36	The BASIC member
BGC	BGC36	The business graphics chart
BGD	BGD36	The business graphics data
BGF	BGF36	The business graphics format
COB	CBL or CBL36	The COBOL member
CSM		Communications and System Management member*
CSP		Cross-system product member*
DFU	DFU36	The data file utility member
DLS		Document library service member*
DTA	DTA36	The data member
FMT	DSPF or DSPF36	The display format member
FOR	FOR36	The FORTRAN member
ICF		CONFIGIC procedure member*
KEY		KEYS procedure member*
MNU	MNU or MNU36	The menu member
MSG	MSGF or MSGF36	The message member
PHL	PHL36	The phone list member
QDE		Query data entry member*
QRY		Query member*
RPG	RPG or RPG36	The RPG member
SRT	SRT36	The sort member
SSP		CNFIGSSP procedure member*
TXT	TXT	Text member
UNS	UNS36	Unspecified
WSU	WSU36	The work station utility member
X25		X.25 packet switching control link*

\* These subtypes are supported for compatibility only. No OS/400 subtype corresponds to these System/36 subtypes. If one of these subtypes were specified, no members would be found.

**MEMBERS**

This specifies listing of the contents of the specified source and procedure members (contents of O and R are not printed). If you do not specify a parameter, MEMBERS is assumed.

**DIRINFO** This specifies listing of only the directory information for the specified member. If you specify DIRINFO without also specifying a member name, it is the same as specifying DIR as the first parameter.

### Example 1

The following example lists the procedure member PAYROLL in the library called MYLIB:

```
LISTLIBR PAYROLL,PROC,MYLIB
```

### Example 2

The following example lists library members that have names beginning with PAY in the library named PAYLIB:

```
LISTLIBR PAY,ALL,LIBRARY,PAYLIB
```

### Example 3

The following example lists only the directory information for all library member types that begin with PAY in a library named PAYLIB:

```
LISTLIBR PAY,ALL,LIBRARY,PAYLIB,,,DIRINFO
```

### Example 4

The following example lists all the members of a library named PAYLIB that are of the subtype COB:

```
LISTLIBR ALL,LIBRARY,PAYLIB,,,COB
```

### Example 5

The following example lists all the source members in the current library and directs the output to a file named MYLIST:

```
LISTLIBR DIR,,,,,,MYLIST
```

## Sample LISTLIBR Listings

The following sections include examples of library listings.

### Library Status Listing

The following line is an example of a library status listing.

```
LIBRARY - TESTLIB          CONTAINS      8      ENTRIES
```

### Library Directory Listing

Figure 4-3 on page 4-152 is an example of a library directory listing.

# LISTLIBR

LIBRARY - TESTLIB		DIRECTORY		DATE 02/08/88		TIME 12.48					
NAME	TYPE	SUBTYPE	DATE	TIME	REF NUMBER	ATTRIBUTES	REL LEV	MRT MAX	PROGRAM SIZE/K BYTES	RECORD LENGTH	NUM STMTS
PGM1	O	CLP	02/08/88	12.39		PGM			10		
PRC1	P	OCL36	01/27/88	10.22	000000					132	1
TEST	S	UNS36	02/05/88	13.32	000000					132	1
MSG5	O		02/08/88	12.38		MSGF					
PRTFILE	O	PRTF	02/08/88	12.42		FILE					
SCREENS	O	DSPF	02/08/88	12.41		FILE					

Figure 4-3. Library Directory Listing

The following is a description of the items on the listing:

**NAME** This specifies the name of the library member.

**TYPE** This specifies the type of library member:

- O** The load member
- P** The procedure member
- R** The subroutine member
- S** The source member

**SUBTYPE**

This specifies the subtype of the member. It can be one of the following System/36 environment subtypes:

Subtype	Meaning
<b>ARS36</b>	ARS, an automatic response member
<b>ASM36</b>	ASM, an assembler member
<b>BASP36</b>	BAP, a BASIC procedure (source member)
<b>BAS36</b>	BAS, a BASIC member
<b>BGC36</b>	BGC, a business graphics chart member
<b>BGD36</b>	BGD, a business graphics data member
<b>BGF36</b>	BGF, a business graphics format member
<b>CBL36</b>	COB, a COBOL member
<b>DFU36</b>	DFU, a data file utility member
<b>DTA36</b>	DTA, a data member
<b>DSPF36</b>	FMT, a display format member
<b>FOR36</b>	FOR, a FORTRAN member
<b>MNU36</b>	MNU, a menu member
<b>MSGF36</b>	MSG, a message member
<b>PHL36</b>	PHL, a phone list member
<b>RPG36</b>	RPG, an RPG member
<b>RPT36</b>	ARP, an RPG auto report member
<b>SRT36</b>	SRT, a sort member
<b>TXT</b>	TXT, a text member
<b>UNS36</b>	UNS, an unspecified member
<b>WSU36</b>	WSU, a work station utility member

The subtype could also be an OS/400 object type attribute such as PRTF, CLP, PF, and so on. For a complete list of these attributes, see the *CL Reference* book.

**DATE** This specifies the date (in system format) the member was either created or last updated.

**TIME** This specifies the time the member was either created or last updated.

**REF NUMBER**

This specifies the reference number assigned to the member. For more information, see "Reference Numbers for Library Members" on page 1-18.

**ATTRIBUTES for USER parameter**

This specifies the various attributes associated with each library member and may be one or more of the following:

- SSP** System-supplied member.
- MRT** A multiple requestor terminal program or procedure.
- NEP** A never-ending program.
- NOLOG** The procedures OCL statements are not to be logged to the history file.
- NONINQ** The program is noninquirable.
- PDATA** The procedure passed data to a program, not parameters. The OS/400 object type may also be listed. See the *CL Reference* book for a complete list of OS/400 object types.

**REL LEV** This is the release level of System/36 environment (IBM-supplied) programs.

**MRT MAX**

This is for load members (O) only. The maximum number of requester display stations that can be attached to the program.

**PROGRAM SIZE**

This is for load members (O) only. The number of kilobytes in the stored program.

**RECORD LENGTH**

This is for source (S) and procedure (P) members only. The length of the statements in the member.

**NUM STMTS**

This is for source (S) and procedure (P) members only. The number of statements in the member. Plus signs indicate that the number of statements is greater than 99 999.

When a listing is printed to the output file, the directory entry fields are in the columns shown in the following table:

Beginning Column	Field Length	Item
1	10	NAME
14	1	TYPE
18	10	SUBTYPE
30	8	DATE
40	5	TIME
47	6	REF NUMBER
55	35	ATTRIBUTES (Columns 55, 62, 69, 76, 83) (Each attribute is 6 bytes.)

Beginning Column	Field Length	Item
91	3	REL LEV
96	3	MRT MAX
101	8	PROGRAM SIZE/K BYTES
115	5	RECORD LENGTH
123	5	NUM STMTS

When a listing is printed to the output file, the library name and number of entries are in the columns shown in the following table:

Line	Beginning Column	Field Length	Item
1	11	10	Library name
1	37	7	Number of entries

### Library Members Listing

Figure 4-4 is an example of a library members listing. The first 12 columns of the listing are not part of the source or procedure member, these are control fields used by the source entry utility (SEU).

```

LIBRARY - TESTLIB                                DATE 02/08/88                                TIME 12.48

  NAME  TYPE  SUBTYPE  DATE  TIME  REF  ATTRIBUTES  REL  MRT  PROGRAM  RECORD  NUM
-----  ---  -----  ----  ----  ---  -----  ---  ---  -----  -----  ---
          P   OCL26   02/08/88 12.49 000000                                132   3
000100880126// * 'THIS IS A TEST PROC'
000200880208// * 'USED TO DISPLAY '
000300880208// * 'THIS PROCEDURE MEMBER'

```

Figure 4-4. Library Members Listing

## LISTNRD Procedure

Not supported. On System/36, the LISTNRD procedure listed either a single entry or all entries from the network resource directory.

On the AS/400 system, the information that was stored in the NRD is kept in OS/400 distributed data management (DDM) files. Each DDM file is the equivalent of one NRD entry.

To do a similar function, use the Display Distributed Data Management File (DSPDDMF) command.

See the *CL Reference* book for more information on the DSPDDMF command. See the *Distributed Data Management* book for more information on using DDM files.

---

## LOAD3601 Procedure

Not supported. On System/36, the LOAD3601 procedure transmitted the contents of a diskette file for the SSP-ICF Finance subsystem from System/36 to the 3601 Finance controller.

On the AS/400 system, use the Send Finance Diskette Image (SNDFNCIMG) command. For more information on using this command, see the *Finance Communications Programming* book.

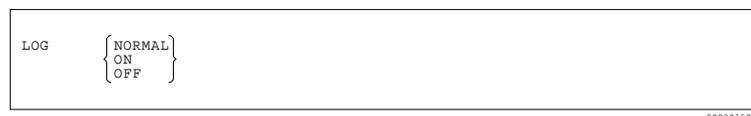
---

## LOG Procedure

The LOG procedure indicates whether to log the OCL statements and utility control statements in a procedure to the job log, regardless of the statement logging indicator in the procedure. This allows you to create your procedures with this indicator set to off, but still have the statements logged to the job log while you are debugging the procedure. The statements that start procedures and end of job information are always logged to the job log.

You can log the options from the system help support menus and from the user menus to the job log by entering LOG ON. This allows you to determine the options that an operator used. If you do not want to log the options, enter LOG NORMAL or LOG OFF.

The LOG statement processes a LOG OCL statement.



**NORMAL** This specifies the using of each procedure's logging indicator.

**ON** This specifies the logging of all OCL statements to the job log, regardless of the procedure's logging indicator. Also, the menu options an operator took are logged.

**OFF** This specifies no logging of OCL statements to the job log, regardless of the procedure's logging indicator. Also, the menu options an operator took are not logged.

### Example

The following example temporarily prevents the OCL statements in procedure PROCA from being logged to the job log:

```

LOG OFF
PROCA
LOG NORMAL

```

---

## LRTRLOAD Procedure

Not supported. On System/36, the LRTRLOAD procedure installed the IBM Token-Ring Network support from diskette. See “Managing Licensed Programs” on page 4-2 for information on installing licensed programs.

---

## LRTRSAVE Procedure

Not supported. On System/36, the LRTRSAVE procedure saved the IBM Token-Ring Network support to diskette. See “Managing Licensed Programs” on page 4-2 for information on saving licensed programs.

---

## MAINTX25 Procedure

Not supported. On System/36, the MAINTX25 procedure allowed you to display information about the X.25 links and to alter certain logical channel and virtual circuit configuration information.

There is no similar function available on the AS/400 system.

---

## MCSCONV Procedure

Not supported. On System/36, the MCSCONV procedure allowed you to convert library source members, library procedure members, data files, text folders, and data dictionaries from the national language version (NLV) hexadecimal representation to the multinational character set (MCS) value, and vice versa.

On the AS/400 system, many functions allow you to define which graphic character set and code page to use. Also, the use of multiple language libraries on a single system is supported.

See the *System/36 Environment Programming* book for more information on multiple language support.

---

## MOVEFLDR Procedure

Not supported. On System/36, the MOVEFLDR procedure allowed you to move a folder from one disk location to another.

On the AS/400 system, this function is not required.

See the *System/36 Environment Programming* book for more information on using folders in the OS/400 System/36 environment. See the *Planning for and Setting Up OfficeVision/400* book for more information on folder management on the AS/400 system.

---

## MSDOWNL Procedure

The MSDOWNL procedure allows a user of a 5550 display to download code to run SNA Japanese 3270PC emulation. See the *SNA Japanese 3270PC Emulation User's Guide*, for more information.

The MSDOWNL procedure runs the Send Japanese 3270PC Emulation Code (SNDEMLIGC) command.

```
MSDOWNL
```

59020650-0

The MSDOWNL procedure has no parameters.

### Example

Download code to run SNA Japanese 3270PC emulation:

```
MSDOWNL
```

---

## MS3270 Procedure

The MS3270 procedure signs a 5550 display station on to SNA Japanese 3270PC emulation. The host line, controller, and device must all be varied on before you can enter this procedure. To set System/36 environment 3270 Device Emulation values for location name and national language, you must use the Change System/36 (CHGS36) command.

See the book, *SNA Japanese 3270PC Emulation User's Guide*, for more information.

```
MS3270 [ location name ] [ ,LU address  
      ,ANY  
      ,NODIRECT ]
```

59020651-0

### location name

This specifies the location name to be associated with this session. The location name, specified when configuring the emulation device description (CRTDEVHOST command), refers to the remote location with which communications take place. When a location value has been added to the special System/36 environment configuration information (CHGS36) command, then this parameter can be omitted.

### LU address

This specifies the LU address to be allocated for the printer session. Host Direct Print should have been selected on customization of 55X0 if this parameter is specified. If Device ID had been specified on configuration for the emulation device description with this LU address, the device ID is ignored. The range of possible values is from hex 01 to hex FE.

### ANY

This specifies that any one of the printer LU addresses, which can be used for 3270 printer session when the MS3270 procedure is run, is allocated. Those LU addresses for which Device ID had been specified

through configuration are omitted from the search. Host Direct Print should have been selected on customization of 55X0 if this parameter is specified.

### **NODIRECT**

This specifies that no LU address is allocated to Host Direct Print. Even if 55X0 had been customized with Host Direct Print, neither LU1 print nor LU3 print is available. Local Copy (both Host Initiated and Operator Initiated) is still available if printer is attached to the 55X0.

The 5550 PC is customized for Japanese emulation when the MSDOWNL procedure is run.

### **Example**

This example signs your 5550 display device on to SNA Japanese 3270PC emulation. This example assumes that a default emulation location name and national language have been set by using the CHGS36 command.

MS3270

---

## **MSGFILE Procedure**

Not supported. On System/36, the MSGFILE procedure allowed you to use the system message file (#MESSAGE). You could do the following:

- Define the size and location of the message file.
- List the display stations and users that have messages in the message file.
- Remove messages from the message file.

On the AS/400 system, messages are sent to message queues. Individual message queues are created for each user and display station. The size of each queue is defined when the queue is created. The size of each queue can only be changed by deleting the queue and creating it again. Message queues can be created to extend automatically when full. If you are authorized to a message queue, you may display or delete messages in the queue.

To do a similar function:

- Use the Create Message Queue (CRTMSGQ) command to create a message queue.
- Use the Delete Message Queue (DLTMSGQ) command to delete a message queue.
- Use the Display Message (DSPMSG) command to display messages in a message queue.

See the *CL Reference* book for more information on the CRTMSGQ, DLTMSGQ, and DSPMSG commands. See the *System Operation* book for more information on messages. See the *System/36 Environment Programming* book for more information on messages in the System/36 environment.

## MSRJE Procedure

The MSRJE procedure starts the remote job entry (RJE) function. For more information about RJE, see the *Remote Job Entry (RJE) Guide*.

On the AS/400 system, the MSRJE procedure does not bring up the interactive reader and console on the same display at the same time. Also, the RJE function cannot submit a batch reader job and then append an interactive reader to the same job.

To read a disk file:

```
MSRJE  location name, [file name], [mmddy
                               ddmmyy
                               yymmdd], [DISK
                                         (D)],
      [COMMAND
      DATA], [DELETE
      NODEL], [NOREL
      RELEASE
      CONSOLE]
```

59020163-0

To read a library member:

```
MSRJE  location name, [member name], [library name
                                     current library], [PROC
                                                         (P)
                                                         SOURCE
                                                         (S)],
      [COMMAND
      DATA], [DELETE
      NODEL], [NOREL
      RELEASE
      CONSOLE]
```

59020164-0

### location name

This specifies the location with which communication is performed. The name is the session name assigned during remote job entry configuration. The session must be active.

**file name** This specifies the disk file to read.

On the AS/400 system, the MSRJE procedure does not call procedures specified in the file parameter.

### member name

This specifies the library member read. The library member can be either a source or procedure member.

### library name

This specifies the library that contains the library member.

### mmddy, ddmmyy, or yymmdd

This specifies the creation date of the disk file. The date, if specified, must be in the session date format. Use the STATUS SESSION command to determine the session date format. If you do not specify this parameter, the most recent disk file is used.

### DISK, PROC, or SOURCE

This specifies whether a disk file or library member is read. DISK is the default.

**DISK or D**

This specifies the name parameter as a disk file.

**PROC or P**

This specifies the name parameter as a library procedure member.

**SOURCE or S**

This specifies the name parameter as a library source member.

**COMMAND or DATA**

This specifies whether the disk file or library member it reads contains only data or both data and control statements. DATA is the default.

**COMMAND**

This specifies that the file or library member contains RJE function utility control statements. The RJE function processes all the RJE function control statements and transmits all other records or statements. If a file is read, the file is processed consecutively. The file or library member must have a record length of 80.

**DATA**

This specifies that the file or library member read contains data. If a file is read, the records process consecutively.

**DELETE or NODEL**

This specifies whether, if a file processes, any deleted records in the file are processed. If the file is not delete-capable, this parameter is ignored. NODEL is the default.

**DELETE**

This specifies transmission of deleted records.

**NODEL**

This specifies no transmission of deleted records are transmitted.

**NOREL**

This specifies that the interactive reader is displayed after RJE function has started. If you do not specify a parameter, NOREL is assumed.

**RELEASE**

This specifies the release of the display station after RJE function has started. This allows you to do other tasks at the display station while the data transmits.

**CONSOLE**

This specifies use of the RJE function console as the display station.

**Example 1**

The following example signs on to RJE function as the RJE function console operator:

```
MSRJE STATION1,,,,,CONSOLE
```

**Example 2**

The following example signs on to RJE function and transmits a library source member that contains both data and process control statements that are contained in the source member. The name of the library member is PAYROLL, and it is contained in the library named MYLIB.

```
MSRJE STATION1,PAYROLL,MYLIB,SOURCE,COMMAND
```

## NOHALT Procedure

The NOHALT procedure specifies the automatic response severity level for the system, your session, or a single job. This allows the system to automatically respond to messages with response values, rather than requiring operator response to an error message.

For more information on automatic response and the severity level, see “RESPONSE Procedure” on page 4-196.

If entered from the keyboard, it remains in effect until another NOHALT procedure is entered, or until the operator signs off the system. If it is specified in a procedure, it remains in effect until another NOHALT procedure processes, or until the job ends.

When setting the automatic response level for MRT procedures, include a // NOHALT x, JOB in the MRT procedure, where x is the severity level required. This insures a consistent severity level for each user that attaches to the MRT program. If a // NOHALT x, JOB statement is not included in the MRT procedure, the severity level is set to zero, and remains so until the procedure ends.

The NOHALT procedure processes a NOHALT OCL statement:

```
NOHALT severity level, [ JOB
                        SESSION
                        SYSTEM ]
```

59020165-0

### severity level

This specifies the automatic response severity level. Enter any of the following values:

- 0** This specifies automatic system response for no messages. If a message displays, the operator must enter a response to the message.
- 1** This specifies automatic system response to any messages having a severity level of 1.
- 2** This specifies automatic system response to any messages having a severity level of 1 or 2.
- 3** This specifies automatic system response to any messages having a severity level of 1, 2, or 3.
- 4** This specifies automatic system response to any messages having a severity level of 1, 2, 3, or 4.

**JOB** This specifies that the severity level affects only the current job containing the NOHALT procedure. If you do not specify a parameter, JOB is assumed. After the job ends, the severity level that is in effect for the session is used.

**SESSION** This specifies that the severity level affects your session (all your jobs until you sign off the system). If the NOHALT procedure is within a procedure and you specify SESSION, the specified level does not take effect until the current job ends. If you do not specify a parameter when

the NOHALT procedure is entered at the keyboard, SESSION is assumed.

**SYSTEM** This specifies that the severity level establishes the system severity level. That means the severity level is established for display stations that sign on *after* the SYSTEM parameter processes. Display stations that were already signed on are not affected.

To establish the system severity level, include the NOHALT procedure in the initial program load (IPL) start-up procedure #STRTUP1. See “#STRTUP1 Procedure” on page 4-3 for more information.

If the NOHALT procedure is within a procedure and you specify SYSTEM, the specified level takes effect immediately for the system automatic response severity level, but does not take effect for the display station session until the current job ends.

An operator must have \*JOBCTL special authority to specify SYSTEM. Refer to the Change User Profile (CHGUSRPRF) command in the *CL Reference* book for a description of this authority.

### Example 1

The following example, placed in the procedure named #STRTUP1, establishes an automatic response severity level of 3 for the entire system:

```
NOHALT 3,SYSTEM
```

### Example 2

The following example, placed in the payroll procedure, establishes an automatic response severity level of 3 for the payroll program. The severity level of 3 affects only the statements that follow the NOHALT procedure. After the procedure ends, the session severity level resumes.

```
NOHALT 3
// LOAD PAYROLL
// FILE NAME-EMPLOY,DISP-OLD
// RUN
```

### Example 3

The following example, entered from your keyboard, establishes an automatic response severity level of 3 for your session:

```
NOHALT 3
```

---

## OFCBPRT Procedure

Not supported. On System/36, the OFCBPRT procedure allowed a user to request batch printing and deletion of calendar items without going through Personal Services/36 menus and displays.

There is no similar function available on the AS/400 system. You can print a calendar interactively by using the Start Office (STROFC) command. To do this, select option 1 (Calendar), then press F15 (Print calendar). You can delete a calendar item interactively by using the STROFC command. To do this, select option 9 (Administration), then select option 5 (Work with office files), then type 1 (Calendar) and 4 (Delete) and press Enter.

See the *Using OfficeVision/400* book for more information on the OfficeVision for OS/400 calendar function.

---

## OFCCAL Procedure

The OFCCAL procedure creates or maintains a calendar using OfficeVision for OS/400. You can create, change, delete, or print a calendar, as well as schedule, reschedule, or cancel calendar items.

The OFCCAL procedure starts the Start Office (STROFC) command, and always displays the five-day calendar or the More Calendar Tasks display, depending on whether the user has a default calendar defined.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.



59020166-0

- WORK** This is supported for System/36 compatibility only. The value is not used.
- PRINT** This is supported for System/36 compatibility only. The value is not used.
- UPDATE** This is supported for System/36 compatibility only. The value is not used.
- MYCAL** This is supported for System/36 compatibility only. The value is not used.
- OTHER** This is supported for System/36 compatibility only. The value is not used.
- CHANGE** This is supported for System/36 compatibility only. The value is not used.
- CREATE** This is supported for System/36 compatibility only. The value is not used.
- DELETE** This is supported for System/36 compatibility only. The value is not used.

---

## OFCCANCL Procedure

Not supported. On System/36, the OFCCANCL procedure stopped Personal Services/36 background mail tasks.

To do a similar function, use the End Subsystem (ENDSBS) command and specify QSNADS for the subsystem (SBS) parameter.

See the *CL Reference* book for more information on the ENDSBS command. See the *SNA Distribution Services* book for more information about SNADS. See the

Using *OfficeVision/400* book for more information on the OfficeVision for OS/400 licensed program.

---

### OFCCOMM Procedure

Not supported. On System/36, the OFCCOMM procedure allowed you to maintain System/36 or Personal Services/36 communications definitions. System/36 communications definitions refer to using advanced program-to-program communications (APPC) to specify information for the Interactive Communications Feature (SSP-ICF) subsystems definitions. Personal Services/36 communications definitions include queue definitions, remote destination definitions, and communications routes.

On the AS/400 system, communications queues are referred to as *distribution queues*.

There is no similar function available on the AS/400 system. To display distribution queues and routing table information use the Configure Distribution Services (CFGDSTSRV) command.

See the *CL Reference* book for more information on the CFGDSTSRV command. See the *System/36 Environment Programming* book, the *SNA Distribution Services* book, and the *APPC Programming* book for more information on the AS/400 communications support. See the *Using OfficeVision/400* book for more information on the OfficeVision for OS/400 licensed program.

---

### OFCCONV Procedure

Not supported. On System/36, the OFCCONV procedure converted user profile data from Release 4 format to Release 5 format.

On the AS/400 system, this function is not required. User profile data migrated from System/36 is not contained in folders on the AS/400 system.

See the book, *System/36 to AS/400 Migration Aid User's Guide and Reference* for more information on the migration of System/36 office user profile data. See the *Using OfficeVision/400* book for more information on the OfficeVision for OS/400 licensed program.

---

### OFCDATA Procedure

Not supported. On System/36, the OFCDATA procedure reorganized or saved office information. Office information consisted of Personal Services/36 files and folders, including calendars and mail logs.

To do a similar function, use the Save Object (SAVOBJ) command, the Save Library (SAVLIB) command, the Save Document Library Object (SAVDLO) command, and the Reorganize Physical File Member (RGZPFM) command.

See the *CL Reference* book for more information on the SAVOBJ, SAVLIB, SAVDLO, and RGZPFM commands. See the *Using OfficeVision/400* book for more information on the OfficeVision for OS/400 licensed program.

---

## OFCDFLT Procedure

The OFCDFLT procedure allows you to change the default values used by Office. You can change the values in your own office profile.

On the AS/400 system, this procedure uses the Start Office (STROFC) command and displays the Administration menu. Select option 1 (Change Enrollment) to work with default values.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.

```

OFCDFLT [ USER
         OFFICE ]
  
```

**USER** This is supported for System/36 compatibility only. The value is not used.

**OFFICE** This is supported for System/36 compatibility only. The value is not used.

---

## OFCDIR Procedure

The OFCDIR procedure selects different ways of looking at the directory.

This procedure uses the Work with Directories (WRKDIR) command and displays the Work with Directories display.

The system distribution directory does not maintain a user name field. However, a user description field displays on the first display along with the user ID. From this display you can sort the list into description sequence.

The system has no phone directory display. You can view telephone numbers by selecting a user ID or description from the list provided by the WRKDIR command.

If you are not a security officer or an office administrator performing the OFCDIR procedure, you do not receive a display of the entire directory. You receive only the details you need.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.

```

OFCDIR [ USER
        GROUP
        NAME
        PHONE ]
  
```

**USER** This is supported for System/36 compatibility only. The value is not used.

- GROUP** This is supported for System/36 compatibility only. The value is not used.
- NAME** This is supported for System/36 compatibility only. The value is not used.
- PHONE** This is supported for System/36 compatibility only. The value is not used.

---

## OFCFILE Procedure

The OFCFILE procedure allows you to file an electronic document in an existing hard copy document in a folder on a remote system.

This procedure uses the Work with Documents (WRKDOC) command and displays the Work with Documents display for the last used folder. The user must specify the folder that contains the document to file and select option 12 (File Remote) for the document.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.



### ELECTRONIC

This is supported for System/36 compatibility only. The value is not used.

### HARDCOPY

This is supported for System/36 compatibility only. The value is not used.

---

## OFCGRP Procedure

The OFCGRP procedure allows you to work with a user group. This procedure uses the Work with Distribution Lists (WRKDSTL) command and displays the Work with Distribution Lists display.

To update a distribution list, you must have proper authority.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.



The OFCGRP procedure has no parameters.

---

## OFCINSTL Procedure

Not supported. On System/36, the OFCINSTL procedure installed Personal Services/36 files on the system.

See the *Using OfficeVision/400* book for more information on the OfficeVision for OS/400 licensed program.

---

## OFCLDF Procedure

The OFCLDF procedure allows you to view, add, delete, and change the descriptions of access codes, document classes, and keywords for each document library. The user must be an administrator to use this function.

This procedure uses the Start Office (STROFC) command and displays the Administration menu. Select option 6 (Work with Library Descriptions file) to change access codes, document classes, and keywords.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.



**ACCESS** This is supported for System/36 compatibility only. The value is not used.

**CLASS** This is supported for System/36 compatibility only. The value is not used.

**KEYWORD** This is supported for System/36 compatibility only. The value is not used.

---

## OFCLOAD Procedure

Not supported. On System/36, the OFCLOAD procedure installed the Personal Services/36 support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## OFCMAIL Procedure

The OFCMAIL procedure works with mail. You can look at the status of all entries in your mail log or just the action items. You can also send mail and look at the status of mail that you have sent.

Office automatically logs any mail that you send or receive through the system, and you can log any printed copy of mail that you send or receive. You can work with a particular user's mail with the OFCMAIL procedure.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.



**STATUS** This specifies that you want to look at the status of your mail. This parameter only displays the incoming mail log. Press F6 on the Work with Mail display to display outgoing mail status.

**ACTION** This parameter displays the Work with Mail display. To get to the Work with Action Items display, press F9. This display only displays incoming action items. To view outgoing mail items, press F6 on the Work with Mail display. To view outgoing action items, press F9 on the Work with Outgoing Mail Status display.

**ALTERLOG**  
This is supported for System/36 compatibility only. The value is not used.

**HARDCOPY**  
This parameter displays the Work with Mail display. Press F13 to display the More Mail Tasks menu, and select option 1 or 2 on that menu to create local or remote hard-copy references.

**MAILOUT**  
This parameter displays the Work with Mail display. Press F6 to display outgoing mail status.

**RECEIVE** This specifies that you want to look at a list of the new mail in your mail log. You can look at the item itself or the memo slip attached to it, change the description of the item, and print, file, send, or delete the item. You can also delete the entry for an item from your mail log.

**REVLOG** This specifies that you want to look at a list of the entries in your mail log, ordered by date with the most recent entries first. This parameter only displays the incoming mail log. Press F6 on the Work with Mail display to display outgoing mail status. You can look at the item itself or the memo slip attached to it, change the description of the item, and print, file, send, or delete the item. You can also delete the entry for an item from your mail log.

**SELECT** This parameter displays the Work with Mail display. If you are authorized to work on behalf of another user (or users), enter the user's ID in the *Working with mail for* prompt and press the Enter key. If you want to see a list of users on whose behalf you are authorized to work, move the cursor to the prompt and press F4. A list of users on whose behalf you are authorized to work displays and you may select a user from the list.

**SEND** This specifies that you want to send a document. You can specify a distribution list, priority, and security classification for the item.

**SENDNOTE**

This specifies that you want to send a note. You can specify a distribution list, subject, reference, confirmed delivery, priority level, security, and the copy list. You can select the distribution list from a list of users or groups. You can also embed an existing word processing document that is stored in a text folder.

---

## OFCMAINT Procedure

Not supported. On System/36, the OFCMAINT procedure maintained office information. You could maintain communications queue definitions, communications routes, or mail folders. You could also list the contents of a mail folder.

On the AS/400 system, communications queues are referred to as *distribution queues*.

There is no similar function available on the AS/400 system. You can display distribution queues and routing table information by using the Configure Distribution Services (CFGDSTSRV) command.

See the *CL Reference* book for more information on the CFGDSTSRV command. See the *System/36 Environment Programming* book, the *SNA Distribution Services* book, and the *APPC Programming* book for more information on the AS/400 communications support. See the *Using OfficeVision/400* book for more information on the OfficeVision for OS/400 licensed program.

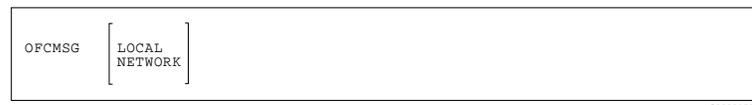
---

## OFCMSG Procedure

The OFCMSG procedure allows you to send messages to a group.

This procedure uses the STROFC command and displays the Send a Message display.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.



**LOCAL** This is supported for System/36 compatibility only. The value is not used.

**NETWORK**

This is supported for System/36 compatibility only. The value is not used.

---

## OFCQ Procedure

Not supported. On System/36, the OFCQ procedure controlled the activity and communications queues.

To do a similar function, use the Work with Document Print Queue (WRKDOCPRTQ) command, and the Work with Distribution Queues (WRKDSTQ) command.

See the *CL Reference* book for more information on the WRKDOCPRTQ and WRKDSTQ commands. See the *Using OfficeVision/400* book for more information on the OfficeVision for OS/400 licensed program.

---

## OFCSAVE Procedure

Not supported. On System/36, the OFCSAVE procedure saved the Personal Services/36 support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## OFCSRCH Procedure

The OFCSRCH procedure works with searches and documents found by a search.

The OFCSRCH procedure uses the Start Office (STROFC) command and displays the Work with Documents and Folders menu. Select option 3 (Search for documents) to request a new search request, or select option 4 (Work with document lists) to use existing search results.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.



The OFCSRCH procedure has no parameters.

---

## OFCSTART Procedure

Not supported. On System/36, the OFCSTART procedure started the background tasks for Personal Services/36.

To do a similar function, use the Start Subsystem (STRSBS) command specifying QSNADS for the subsystem (SBS) parameter.

See the *CL Reference* book for more information on the STRSBS command. See the *SNA Distribution Services* book for more information about SNADS. See the *Using OfficeVision/400* book for more information on the OfficeVision for OS/400 licensed program.

---

## OFCSTAT Procedure

The OFCSTAT procedure allows you to view a list of document library requests. Document library requests are the functions that can be carried out using document library services such as filing, searching for, and retrieving documents.

This procedure uses the Start Office (STROFC) command and displays the Work with Documents and Folders menu. Select option 5 (Work with remote status) to view document library requests.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.

```
OFCSTAT  { REQUESTS }
```

S90205B2-0

### REQUESTS

This is supported for System/36 compatibility only. The value is not used.

---

## OFCUSER Procedure

The OFCUSER procedure enrolls or changes the enrollment of general and indirect users of AS/400 Office. An indirect user is not enrolled. The user must be an administrator to enroll new users or change enrollment for any users other than himself.

This procedure uses the Start Office (STROFC) command and displays the Administration menu. Select option 1 (Create and maintain enrollment) to perform this function.

If the procedure returns without displaying a display or an error message, use the Display Job Log (DSPJOBLOG) command and press F10 (Display detailed messages) for more information.

```
OFCUSER  { ENROLL } , { OFC }
          { CHANGE } { IWS }
```

S9020178-1

**ENROLL** This is supported for System/36 compatibility only. The value is not used.

**CHANGE** This is supported for System/36 compatibility only. The value is not used.

**OFC** This is supported for System/36 compatibility only. The value is not used.

**IWS** This is supported for System/36 compatibility only. The value is not used.

### **OLINK Procedure**

Not supported. On System/36, the OLINK procedure link-edited compiled programs and subroutines to create a library load member that could be run.

On the AS/400 system, this function is not required. Compiled programs do not have to be link-edited before they can be run. See "LOAD OCL Statement" on page 5-55 and "RUN OCL Statement" on page 5-83 for information about running programs.

---

### **OLPDLOAD Procedure**

Not supported. On System/36, the OLPDLOAD procedure installed the online problem determination (OLPD) support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

### **OLPDSAVE Procedure**

Not supported. On System/36, the OLPDSAVE procedure saved the online problem determination (OLPD) support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

### **ORGANIZE Procedure**

The ORGANIZE procedure is supported only for compatibility with the IBM System/34. To organize a disk file, see "COPYDATA Procedure" on page 4-50. To organize a disk file and save it on diskette, see "SAVE Procedure" on page 4-220.

The ORGANIZE procedure can also be used with physical, database data files that do not have a System/36 environment direct, indexed or sequential file organization. When such files are used, the new file may be created with file attributes that differ from those you expect. A message allowing you to continue is sent whenever this may occur. See the *System/36 Environment Programming* book for more information on using System/36 environment files and AS/400 files.

---

### **OVERRIDE Procedure**

The IBM System/34 OVERRIDE procedure is not supported. On System/36, the user was referred to the ALTERCOM procedure.

Similar function is available from the Configure Devices and Communications (CFGDEVCMN) menu. From this menu you can change communications configuration information.

See the *Local Device Configuration* book for more information on communications and device configuration.

## PASSTHRU Procedure

Not supported. On System/36, the PASSTHRU procedure allowed you to pass through from your system to a remote System/36 or System/38 where you could sign on as if your display station were attached to the remote system.

To do a similar function, use the Start Pass-Through (STRPASTHR) command. The following is a mapping of System/36 PASSTHRU procedure parameters to STRPASTHR command parameters:

```
PASSTHRU  remote location name, [session group name], [node list member name],
          [node list member library], [virtual control unit name]
```

59020510-2

### remote location name

This specified the remote location that you wanted to sign on to. This maps to the RMTLOCNAME parameter on the STRPASTHR command.

### session group name

This specified the APPC session group used by display station pass-through. The OS/400 object which corresponds to a System/36 session group is a **mode**. This parameter maps to the MODE parameter on the STRPASTHR command.

Like System/36, the mode (session group) parameter is not required. If it is not specified, the system's default mode is used.

### node list member and library names

This specified the name and library of a source member that contained a list of the System/38 location names that acted as intermediate nodes in order to complete the pass-through connection. If this parameter was not specified, the display station pass-through support assumed that no System/38 system was needed to act as an intermediate node to complete the pass-through session. On the AS/400 system, you must specify this information on the CNNDEV parameter. The STRPASTHR command does not support getting this information from a source member. Each System/38 or APPC device name must be specified explicitly on the CNNDEV parameter.

On System/36, a remote location name was specified when a node list member was specified. This remote location name instructed the pass-through software to make a connection to that location name (presumably an intermediate System/38) before using the node list to complete the path to the remote system. There is no similar function on the AS/400 system. If you specify a list of APPC devices on the CNNDEV parameter, you are not allowed to specify a location name for the RMTLOCNAME parameter.

### virtual control unit name

This specified the name of the virtual control unit that was created using the CRTCUD command on the System/38. This prompt was required if the remote system was a System/38.

## PASSWORD

On the AS/400 system, you must specify this information on the VRTCTL parameter. If the remote system is another AS/400 system, this parameter specifies a virtual controller name (created by using the CRTCTLVWS command). If the remote system is a System/38, this parameter specifies a virtual control unit name (created by using the CRTAUD command). You may request automatic configuration of virtual controllers (VRTCTL) and virtual devices (VRTDEV) by not specifying the VRTCTL and VRTDEV parameters on the Start Pass Through (STRPASTHR) command. If you want to use one virtual device that is a part of a specific group of virtual devices, either specify the VRTCTL to which all the virtual devices are attached using the VTRCTL parameter, or specify each of the virtual devices by name on the VRTDEV parameter.

There are additional parameters on the STRPASTHR command that provide functions that were not available on the PASSTHRU procedure.

### Ending Display Station Pass-Through

When you are passed-through to a System/36, the pass-through session is ended by signing off the System/36. When you are passed-through to a System/38 or another AS/400 system, the pass-through session is ended by using the End Pass-Through (ENDPASTHR) command.

### System/36 Display Station ID

On the System/36, a display station ID unique to the remote System/36 was generated when the pass-through session was started, and you could see that ID in the upper right corner of your display.

On the AS/400 system, a virtual display device description is associated with each pass-through session on the remote system. The virtual display device, and its associated virtual controller, must exist before the pass-through session can be established. The name of the virtual device is displayed on the sign-on screen. For users running in the OS/400 System/36 environment, the virtual device name is converted to a System/36-style two-character display ID. This two-character display ID is shown on most System/36 environment user menus, and you can determine the name of the corresponding virtual device with the Display System/36 (DSPS36) command. You can also determine the name of the device after you sign on by using the Display Job (DSPJOB) command or the Work with Active Jobs (WRKACTJOB) command.

See the *CL Reference* book for more information on the STRPASTHR, ENDPASTHR, DSPS36, DSPJOB, and WRKACTJOB commands. See the *Local Device Configuration* book for more information on controller and device description objects.

---

## PASSWORD Procedure

The PASSWORD procedure allows you to change your password.

For more information about password security, see the *System/36 Environment Programming* and the *Security – Reference* book.

The PASSWORD procedure runs the \$PRPWD utility program.

```
PASSWORD [CHANGE]
```

59020511-1

**CHANGE** This specifies changing of your password. You do not need to specify CHANGE since it is the default.

### Example

The following example prompts you for a new password and changes your password:

```
PASSWORD
```

---

## PATCH Procedure

Not supported. On System/36, the PATCH procedure displayed selected disk or diskette sectors and allowed you to change the data in those sectors.

There is no similar function available on the AS/400 system. The way the AS/400 system manages disk storage is different from System/36. You can dump the disk storage for an object by using the Dump Object (DMPOBJ) or Dump System Object (DMPSYSOBJ) commands. You can display and change disk storage by running the *Display/Alter/Dump* service function provided by the Start System Service Tools (STRSST) command.

See the *CL Reference* book for more information on the DMPOBJ, DMPSYSOBJ, and STRSST commands. See the *System Operation* book for more information on working with system problems. See the *AS/400 Licensed Internal Code Diagnostic Aids – Volume 1* book for more information on the system service tools (SST).

---

## PCEXCH Procedure

Not supported. On System/36, the PCEXCH procedure was part of the PC Support/36 and the File Support Utility program products. The PCEXCH procedure allowed you to exchange data between a virtual disk or virtual diskette and a folder. The File Support Utility could create a virtual diskette on the System/36; PC Support/36 could create a virtual disk. Virtual disks or virtual diskettes contained data from an IBM Personal Computer.

On the AS/400 system, virtual disks are not supported. You can use the Convert To Folder (CVTTOFLR) command to convert a virtual disk to a shared folder. See “PCU Procedure” on page 4-177 for information on how to use shared folders.

See the *Client Access/400 for DOS with Extended Memory User Guide* book and the *Client Access/400 for OS/2 User Guide* book for more information on Client Access for OS/400.

---

## PCEXEC Procedure

The PCEXEC procedure allows the Client Access for OS/400 organizer user to issue commands on the personal computer.

If the Client Access for OS/400 organizer is not active and you use the PCEXEC procedure, a message appears telling you that PCO.EXE is not active. Therefore, there can be no communications with the personal computer. Press the Enter key to resume.

```
PCEXEC  PC command, [ PAUSE
                    ]
                    ]
```

59020675-0

### PC command

This specifies the command you want to execute on the personal computer. If you enter COMMAND.COM, the personal computer environment is assumed, and you remain in the personal computer until you type EXIT to return to the System/36 environment. If you enter a personal computer command containing either blanks or commas, enclose the personal computer command in single quotes ("").

**PAUSE** This specifies that you can press any key to return after the personal computer command has run. If the first parameter COMMAND.COM, this parameter must be blank or an error is issued.

### NOPAUSE

This specifies that the Client Access for OS/400 organizer returns to the System/36 environment immediately after the personal computer command has run. If the parameter is COMMAND.COM, this parameter must be blank or an error is issued.

### Example

The following example runs a DIR command on the personal computer while running under the Client Access for OS/400 organizer. The personal computer display remains on the screen until you press a key to return to the Client Access for OS/400 organizer.

```
PCEXEC  'DIR /W',PAUSE
```

---

## PCOLOAD Procedure

Not supported. On System/36, the PCOLOAD procedure installed the PC Support/36 Organizer from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## PCOPROF Procedure

The PCOPROF procedure allows you to select a text editor. The PCOPROF procedure takes information from the displays and stores that information in a profile. If there is no profile, the system creates one for you.

```
PCOPROF
```

59020594-0

The PCOPROF procedure has no parameters.

---

## PCOSAVE Procedure

Not supported. On System/36, the PCOSAVE procedure saved the PC Support/36 Organizer to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## PCU Procedure

The personal computer utility (PCU) procedure allows you to do the following:

- Copy all of the PC files within a virtual disk or diskette to a shared folder.

Document types with PC file extensions of:

- RFT will default to RFTDCA (revisable form text document content architecture).
- FFT will default to FFTDCA (final form text document content architecture).
- All others will default to PCDATA.
- Use Client Access for OS/400.
- All other functions supported by the PCU procedure on System/36 are not supported.

To copy all PC files within a virtual disk or diskette to a shared folder, do the following:

```
PCU  SHRFLDR,DISKCOPY,[disk/diskette name],[folder name],[NOREPLACE  
REPLACE]
```

59020595-0

### DISKCOPY

This specifies that you want to copy all of the files and directories from the virtual disk to the folder.

**disk** This specifies the name of the virtual disk copied.

**diskette** This specifies the name of the virtual diskette copied.

### folder name

This specifies the name of the folder all of the PC files are copied into.

### NOREPLACE

This specifies that documents with the same name are not replaced with the PC file information.

### REPLACE

This specifies that existing documents with the same name as the PC files copied are replaced with the PC file.

### Example 1

The following example starts the PCU procedure to use Client Access for OS/400:

```
PCU  
or  
PCU SHRFLDR
```

A menu appears. Select one of the options from that menu.

### Example 2

The following example converts a virtual disk (DISKA) to a folder (FOLDERA):

```
PCU SHRFLDR,DISKCOPY,DISKA,FOLDERA
```

---

## POST Procedure

Not supported. On System/36, the POST procedure copied special E-format diskette files to disk files. The special E-format diskettes were created by the IBM 5260 Retail System. The POST procedure could also:

- Copy a special E-format diskette file or basic data exchange diskette file to a sequential or indexed disk file.
- Add a special E-format diskette file or basic data exchange diskette file to an existing sequential disk file.
- Copy a disk file to a basic data exchange diskette file.
- Add records from a disk file to an existing basic data exchange diskette file.

To do similar functions, with the exception of adding to an existing diskette file, use the Copy From Diskette (CPYFRMDKT) and Copy To Diskette (CPYTODKT) commands, or the Copy File (CPYF) command. These commands can be selected from the Copy Commands (CMDCPY) menu.

See the *CL Reference* book for more information on the CPYFRMDKT, CPYTODKT, and CPYF commands.

---

## PRINT Procedure

The PRINT procedure specifies the following information about the printer used during a display station session unless a specific printer ID is specified in a job:

- The printer ID of the printer used for all printed output created during the display station session
- The number of lines printed per page
- The vertical print density (lines per inch)
- The horizontal print density (characters per inch)
- The forms number
- The orientation or size of printed output on the page (rotation or reduction)
- The printer drawer from which paper is used

A job placed on the job queue uses the values in effect when the job was placed on the queue. However, if a procedure running from the job queue contains a PRINT procedure, the job uses the PRINT procedure in that procedure.

The PRINT procedure processes a FORMS OCL statement.

```

PRINT [printer id], [lines per page], [lpi value], [cpi value],
      [forms number], [degree of rotation], [paper drawer]

```

59020184-1

Although all parameters are optional, you must specify at least one parameter. Any omitted parameters cause the corresponding settings to remain unchanged.

### printer id or SYSTEM

This specifies the printer used for printed output. All system list output, Print key output, and any other printed output from the session prints on the specified printer. You can use the PRINTER OCL statement to specify a different printer for program output. If you do not specify a parameter, the printers are not changed.

**printer id** This specifies the work station ID of the printer used. Use the STATUS WORKSTN command to determine the printer IDs.

**SYSTEM** This specifies use of the system printer.

### lines per page

This specifies the number of print lines per page. Specify a maximum of 112 lines per page. Determine this number by multiplying the lines per inch setting times the height of the paper. For example, if the paper is 11 inches high and the lines per inch setting is 6, the lines per page setting is 66 ( $11 \times 6 = 66$ ). The lines per page remains in effect until either the session ends, or a new lines per page value is specified by the following:

- Another PRINT procedure
- The LINES procedure
- A PRINTER OCL statement (for that print file only)
- A FORMS OCL statement
- The SET procedure or the \$SETCF utility program

If you do not specify a parameter and the number of lines per page has not been previously set during the session, the system uses the value specified for the display station during system configuration, by the SET procedure, or by the \$SETCF utility program.

For most user-written programs and System/36 environment utility programs that use the system list device for output, the following is true:

- Printing starts on line 6 of the first page and on line 6 of all new pages.
- If the value specified for LINES is greater than 12, printing skips to a new page when six less than the number of lines specified is printed.

For example, if you specify LINES-66, printing starts on line 6 and skips to a new page after printing line 60. Therefore, you get 55 lines of output per page

## PRINT

(66 – 11 = 55, 5 blank lines at the top of a page and 6 blank lines at the bottom). If you specify LINES-13, two lines are printed per page. When you specify 12 or fewer lines per page, printing occurs on every line of the page.

For System/36 environment utility programs that use OS/400 print files for output, the following is true:

- Printing starts on line 1 of the first page and on line 1 of all new pages.
- If the value specified for LINES is greater than 12, printing skips to a new page when six less than the number of lines specified is printed.

For example, if you specify LINES-66, printing starts on line 1 and skips to a new page after printing line 60. Therefore, you get 60 lines of output per page (66 – 6 = 60, 6 blank lines at the bottom of the page). If you specify LINES-13, seven lines are printed per page. When you specify 12 or fewer lines per page, printing occurs on every line of the page.

**Note:** For a list of the System/36 environment utilities that use the system list device and the OS/400 print files, see the *System/36 Environment Programming* book.

For print operations from your programs, the OS/400 program indicates an overflow condition when six less than the number of lines specified (either in the program or in the PRINT procedure) are printed.

**lpi value** This specifies the vertical print density (that is, lines per inch) to use for printed output from the display station session. If the output prints on a printer that does not support vertical print density, the system ignores the lines per inch value.

The lines per inch value remains in effect until either the session ends or until one of the following specifies a new lines per inch value:

- Another PRINT procedure
- The LINES procedure
- A PRINTER OCL statement (for that print file only)
- A FORMS OCL statement

If you do not specify a lines per inch value and the lines per inch was not previously set during the session, the system uses the value that was set at printer configuration.

**cpi value** This specifies the horizontal print density (that is, characters per inch) used for printed output from the display station session. Specify 10 or 15. The printer you use must support the horizontal print density you specify. If the value you specify is not supported by the printer, a message appears and the operator controlling the printer either continues or cancels the job.

The characters per inch value remains in effect until either the session ends or one of the following specifies a new characters per inch value:

- Another PRINT procedure
- The LINES procedure
- A PRINTER OCL statement (for that print file only)
- A FORMS OCL statement

If you do not specify a characters per inch value and the characters per inch was not previously set during the session, the system uses the value set at printer configuration (either 10 or 15).

### forms number

This specifies the form number of the printer forms used for printed output from the display station session. The FORMSNO parameter is valid only when SPOOL is YES on the PRINTER OCL statement. (Each type of form should have a unique, user-assigned form number.) The form number can be any combination of up to 4 characters, except commas (,), apostrophes ('), or blanks.

**Note:** Use question marks (?), slashes (/), equal signs (=), greater than signs (>), plus signs (+), and hyphens (-) with caution because they have special meanings within procedures.

If you specify a forms number, the operator controlling the printer receives a prompt to install the forms with the specified forms number in the printer if the specified forms are not already installed.

The forms number remains in effect until either the session ends or any of the following specify a new forms number:

- Another PRINT procedure
- The LINES procedure
- A PRINTER OCL statement (for that print file only)
- A FORMS OCL statement

### degree of rotation

This specifies the degree of rotation that output is printed on the page, or that the output is reduced for printing on 8-1/2 x 11-inch paper. Specify 0, 90, 180, or 270 degrees of rotation, or specify COR for computer output reduction. If you specify COR, output normally printed on 14-inch wide paper prints on 8-1/2 x 11-inch paper. This parameter is valid only for output printed on a printer that supports rotation.

The **degree of rotation** value remains in effect until either the session ends or any of the following specify a new **degree of rotation**:

- Another PRINT procedure
- A PRINTER OCL statement ROTATE keyword parameter (for that print file only)
- A FORMS OCL statement ROTATE keyword parameter

## paper drawer

This specifies the printer drawer from which paper is used. Specify 1, 2, or 3 (for drawer 1, drawer 2, or drawer 3).

The **paper drawer** value remains in effect until either the session ends or any of the following specify a new **paper drawer**:

- Another PRINT procedure
- A PRINTER OCL statement DRAWER keyword parameter (for that print file only)
- A FORMS OCL statement DRAWER keyword parameter

### Example 1

The following example specifies that printed output during the session prints on the printer with a printer ID of P3:

```
PRINT P3
```

### Example 2

The following example specifies that the number of lines per page to use is 51:

```
PRINT ,51
```

### Example 3

The following example specifies use of a printer with a printer ID of P2 for printed output during the session. The operator wants the printed output to print at 8 lines per inch and 15 characters per inch. Because the forms installed in the printer are 11 inches (27.9 cm) long, the number of lines per page is 88.

```
PRINT P2,88,8,15
```

---

## PRINTKEY Procedure

The PRINTKEY procedure allows you to specify the following:

- Whether a border should be printed around the display image. This helps in documenting your displays.
- Whether a heading should be printed before the display image. This helps determine where the print request came from when no separator pages are being printed.

The changes made with the PRINTKEY procedure remain in affect during the session until changed by another PRINTKEY procedure or a WORKSTN OCL statement.

The PRINTKEY procedure runs the \$SETCF utility program.

```
PRINTKEY [ printer id ] , [ BORDER NOBORDER ] , [ HEADER NOHEADER ]
```

RV2P181-0

Although all parameters are optional, at least one parameter must be specified. Any parameters that are omitted cause the setting to remain unchanged.

<b>printer id</b>	This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. Print key output is always sent to the printer defined as the session printer.
<b>BORDER</b>	This specifies that a border should be printed around the display image.
<b>NOBORDER</b>	This specifies that no border should be printed around the display image.
<b>HEADER</b>	This specifies that a heading should be printed above the display image.
<b>NOHEADER</b>	This specifies that no heading should be printed above the display image.

### Example 1

The following example shows how to get a border printed around the display when you press the Print key.

```
PRINTKEY ,BORDER
```

### Example 2

The following example shows how to get a border and a header printed. The printer ID P4 is checked for syntax and ignored.

```
PRINTKEY P4,BORDER,HEADER
```

The following example shows a sample Print key output with both a header and border.

# PRINTKEY

---

\*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8  
Print Key Output Page 1  
RCH38333 09/13/94 10:44:27

5763SS1 V3R1M0 940909

Display Device . . . . .: WG  
User . . . . .: USER7

\*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8  
\*\*\*\*\*

01 \* MAIN AS/400 Main Menu \* 01  
02 \* System: RCH38333 \* 02  
03 \* Select one of the following: \* 03  
04 \* \* 04  
05 \* 1. User tasks \* 05  
06 \* 2. Office tasks \* 06  
07 \* 3. General system tasks \* 07  
08 \* 4. Files, libraries, and folders \* 08  
09 \* 5. Programming \* 09  
10 \* 6. Communications \* 10  
11 \* 7. Define or change the system \* 11  
12 \* 8. Problem handling \* 12  
13 \* 9. Display a menu \* 13  
14 \* 10. User support and education \* 14  
15 \* 11. PC Support tasks \* 15  
16 \* \* 16  
17 \* 90. Sign off \* 17  
18 \* \* 18  
19 \* Selection or command \* 19  
20 \* ==> \* 20  
21 \* \* 21  
22 \* F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=User support \* 22  
23 \* F23=Set initial menu \* 23  
24 \* \* 24

\*\*\*\*\*  
\*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8

---

The following figure shows a sample Print key output without a header or border.

---

```

MAIN                               AS/400 Main Menu
                                     System:   RCH38333
Select one of the following:

    1. User tasks
    2. Office tasks
    3. General system tasks
    4. Files, libraries, and folders
    5. Programming
    6. Communications
    7. Define or change the system
    8. Problem handling
    9. Display a menu
   10. User support and education
   11. PC Support tasks

    90. Sign off

Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=User support
F23=Set initial menu

```

---



---

## PROBLEM Procedure

Not supported. On System/36, the PROBLEM procedure started the online problem determination (OLPD) procedures.

A similar function is available on the Problem Handling (PROBLEM) menu.

See the *System Operation* book for more information on working with system problems.

---

## PROFLOAD Procedure

Not supported. On System/36, the PROFLOAD procedure installed the PROFS\* bridge support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## PROFSAVE Procedure

Not supported. On System/36, the PROFSAVE procedure saved the PROFS bridge support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

## PRTGRAPH Procedure

The PRTGRAPH procedure prints graphic files on an intelligent printer data stream (IPDS) printer. Examples of graphics files are Business Graphics Utilities/36 (BGU/36) graphics files or graphics files built with the BLDGRAPH procedure.

The PRTGRAPH procedure runs the \$DPGP utility.

See the *System/36 Environment Programming* book for more information on creating graphics files and merging text and graphics.

To print the graphics data from a disk file:

```
PRTGRAPH prtId,FILE,graphics file name,width
```

59020557-1

**prtId** This specifies the ID of the IPDS printer on which the graphics file prints. The default is the current session printer.

**FILE** This specifies that the graphics data is in a disk file.

**graphics file name**

This specifies the name of the graphics object file to be printed.

**width** This specifies the width, in inches, of the graphics area to be printed. This parameter cannot contain more than 5 characters specified in decimal numbers. The specified value cannot be more than 45.50. For example, if the area to be printed is 13 inches, you can specify either 13 or 13.0 for this parameter. If you do not specify this parameter, 13.2 is assumed.

To print the graphics data from a library source member:

```
PRTFRAPH prtId,SOURCE,member name,library name
```

59020652-0

**Note:** On System/36, printing a graphic from a source member was only supported by the Intelligent Printer Data Stream Advanced Functions PRPQ.

**prtId** This specifies the ID of the IPDS printer on which the graphics file prints. The default is the current session printer.

**SOURCE** This specifies that the data is in a library source member.

**member name**

The name of the library source member which contains the graphic options and parameters. Each record in the source member must be in the following format:

- Columns 1 through 8, the option name
- Column 9, leave blank
- Columns 10 through 89, the parameters for the option

**Note:** Comment records are supported by this utility. Place an asterisk (\*) in column 1 of the source records containing the comment.

**library name**

The library name which contains the source member. If this value is not specified, only the current library is searched.

---

**PTF Procedure**

Not supported. On System/36, the PTF procedure allowed you to work with program temporary fixes (PTFs).

To do a similar function, use the Load Program Temporary Fix (LODPTF), Apply Program Temporary Fix (APYPTF), Display Program Temporary Fix (DSPPTF), and Remove Program Temporary Fix (RMVPTF) commands. These commands can be selected from the Program Temporary Fix Commands (CMDPTF) menu.

See the *CL Reference* book for more information on the LODPTF, APYPTF, DSPPTF, and RMVPTF commands. See the *System Operation* book for more information on working with system problems.

---

**QRY Procedure**

The QRY procedure uses the Work with Queries (WRKQRY) command to allow you to produce a variety of reports using data from files. The query itself is a question, and the report created from the query is the answer to the question.

You can use QRY to:

- Create a new query, and revise, copy, delete, browse, and run an existing query
- Display or print a query report
- Print a query definition
- Select and sort the data in the report and perform arithmetic operations on it
- Write the query output to a file

For more information about Query/400\*, see the *Query/400 Use* book.

QRY

59020187-0

The QRY procedure has no parameters.

---

**QRYDE Procedure**

The QRYDE procedure allows you to enter new data in a file, or update existing data in a file one record at a time.

QRYDE file name,  $\left[ \begin{array}{l} \text{mddy} \\ \text{ddm} \\ \text{yy} \end{array} \right]$

59020585-0

**file name** This specifies the name of the disk file you want to update or in which you want to enter data. The file you specify must be linked to an IDDU file definition. See the “IDDULINK Procedure” on page 4-116 for more information.

**mmddy, ddmmy, or yymmdd**

This specifies the creation date of the file to update. Use this date when date-differentiated versions of the same file exist.

**Note:** The date specified must be in the same format as the session date. If a date is not specified, and more than one file exists with the same name, the last file created with this name is used.

## QRYLOAD Procedure

Not supported. On System/36, the QRYLOAD procedure installed the Query/36 support from diskette. See “Managing Licensed Programs” on page 4-2 for information on installing licensed programs.

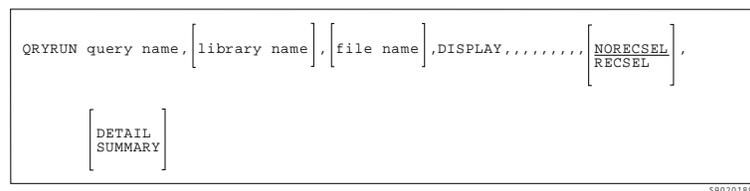
## QRYRUN Procedure

The QRYRUN procedure runs a query that has already been defined and sends the report produced to the device specified. You can also display data in a file without defining a query. Also, you can use the Run Query (RUNQRY) command.

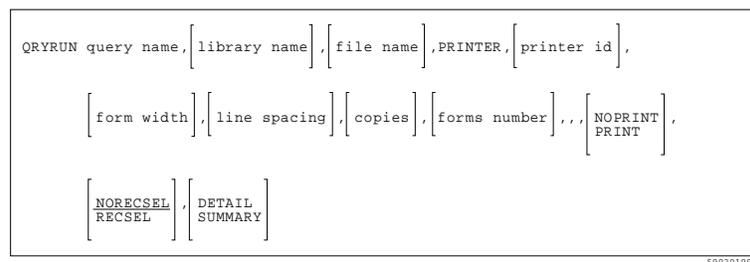
For more information about Query/400, see the *Query/400 Use* book.

The following syntax diagrams demonstrate how to run various queries.

To run a query and display the report:



To run a query and print the report:



To run a query and send the output to disk:

```
QRYRUN query name, [library name], [file name], DISK,,,,, [output file name],
[NEW REPLACE], [NOPRINT PRINT], [NORECSEL RECSEL], [DETAIL SUMMARY]
```

S9020191-3

To display data in a file without defining a query:

```
QRYRUN ,,file name, DISPLAY,,,,,, [NORECSEL RECSEL], [DETAIL SUMMARY]
```

S9020484-2

To print a report without defining a query:

```
QRYRUN ,,file name, PRINTER, [printer id], [form width], [line spacing], [copies],
[forms number],,, [NOPRINT PRINT], [NORECSEL RECSEL], [DETAIL SUMMARY]
```

S9020512-2

To send output to disk without defining a query:

```
QRYRUN ,,file name, DISK,,,,, [output file name], [NEW REPLACE], [NOPRINT PRINT], [NORECSEL RECSEL],
[DETAIL SUMMARY]
```

S9020513-3

**query name**

This specifies the query to run. The query must be previously defined. If you do not specify a parameter, specify a file name.

**library name**

This specifies the library that contains the query to be run. If you do not specify a parameter, the current library is assumed.

**file name**

This specifies the disk file to query. If you do not specify a parameter, the file that was specified when the query was defined is assumed. If you specify a file name but do not specify a query name, the data in the file is displayed, printed, or sent to disk.

**DISPLAY, PRINTER, or DISK**

This specifies where to send the report or output produced by the query. If you do not specify a parameter, the output device that you specified when the query was defined is assumed. If you do not specify an output device in the query, or if you do not specify a query name, DISPLAY is assumed.

**DISPLAY**

This specifies that the report or output produced by the query is sent to the display station that runs the procedure.

**PRINTER** This specifies that the report or output produced by the query prints.

**DISK** This specifies that the output produced by the query is written to a disk file.

**printer id** This specifies the work station ID of the printer on which the report is printed. If you not specify a parameter, the printer that you specified when the query was defined is assumed. If you do not specify a printer in the query, or if you do not specify a query name, the session printer is assumed.

**form width**

This specifies the width of the forms on which the report is printed. Enter a number from 1 to 198. If you do not specify a parameter, the form width that you specified in the query is assumed. If you do not specify a form width in the query, or if you do not specify a query name, 132 is assumed.

**line spacing**

This specifies the number of blank lines to leave between lines in the report. Enter a number from 1 to 3:

- 1 indicates single spacing (no blank lines).
- 2 indicates double spacing (1 blank line).
- 3 indicates triple spacing (2 blank lines).

If you do not specify a parameter, the line spacing specified in the query is assumed. If you do not specify a line spacing in the query, or if you do not specify a query name, 1 is assumed.

**copies** This specifies the number of copies of the report to be printed. Enter a number from 1 to 255. If you do not specify a parameter, the number of copies that you specified in the query is assumed. If you do not specify a number in the query, or if you do not specify a query name, 1 is assumed.

**forms number**

This specifies the name of the form on which you want the report to be printed. Enter from 1 to 4 characters. When the report is ready to print, a message at the subconsole asks if that form is on the printer. Check the printer and change forms (if necessary) before printing.

The forms number cannot contain: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), equal signs (=), and hyphens (-). If you do not specify a parameter, the forms number that you specify in the query is assumed. If you do not specify a number in the query, or if you do not specify a query name, no number is assumed.

**output file name**

This specifies the disk file to contain the query output. If you do not specify a parameter, the file that you specify in the query is assumed. If you do not specify a file in the query and the query output was sent to disk, QQRYOUT is assumed. If you do not specify a query name, no file is assumed.

**NEW or REPLACE**

This specifies whether the report is written to a new disk file or replaces the contents of an existing file. If you do not specify a parameter, the

NEW or REPLACE option that you specified in the query is assumed. If you do not specify an option in the query, or if you do not specify a query name, NEW is assumed.

**NEW** This specifies that the report is written to a new disk file.

**REPLACE**

This specifies that the report replaces the contents of a disk file that already exists on the system.

**NOPRINT or PRINT**

This specifies whether the query definition prints when the query runs. If you do not specify a parameter, the NOPRINT or PRINT option that was specified in the query is assumed. If you do not specify an option in the query, or if you do not specify a query name, NOPRINT is assumed.

**NOPRINT** This specifies that the query definition does not print when the query runs.

**PRINT** This specifies that the query definition prints when the query runs. Do not specify PRINT if DISPLAY is the fourth parameter.

**NORECSEL or RECSEL**

This specifies whether the query runs with a run time selection test. If you do not specify a parameter, NORECSEL is assumed.

**NORECSEL**

This specifies that the query runs without a run time record selection test.

**RECSEL** This specifies that record selection tests change for this run only. A display appears on which you can change the record selection tests defined in the query or specify record selection tests if you did not specify a query.

**DETAIL or SUMMARY**

This specifies the type of output produced by the query. If you do not specify a parameter, the DETAIL or SUMMARY option that you specified in the query is assumed. If you do not specify an option in the query, or if you do not specify a query name, DETAIL is assumed.

**DETAIL** This specifies that the output produced by the query is a report containing detail records and summary records, if any exist.

**SUMMARY**

This specifies that the output produced by the query is a report containing summary records only.

---

**QRYSAVE Procedure**

Not supported. On System/36, the QRYSAVE procedure saved the Query/36 support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## READINFO Procedure

The READINFO procedure allows you to display help text information as a document in its final form. The document is created using the word processing function of the OfficeVision for OS/400 licensed program.

```
READINFO document name, folder name, [ help text label ]
```

59020193-0

**document name**

This specifies the name of the help document you want displayed.

**folder name**

This specifies the name of the folder that contains the help document to be displayed.

**help text label**

This specifies where to start displaying the document. If no label is specified, the document will be displayed at the start of the first page.

**Note:** If the document name parameter or the folder name parameter is not specified, the READINFO prompt display will be shown.

---

## REBLD Procedure

The AS/400 system supports the REBLD procedure for System/34 compatibility only. (The ninth parameter is not supported.) See "COPYDATA Procedure" on page 4-50, for information on copying disk files.

The REBLD procedure can also be used with physical, database data files that do not have a System/36 environment direct, indexed, or sequential file organization. When such files are used, the new file may be created with file attributes that differ from those you requested or expected. A message allowing you to continue is sent whenever this may occur. See the *System/36 Environment Programming* book for more information on using System/36 environment files and AS/400 files.

---

## RELOAD Procedure

The IBM System/34 RELOAD procedure is not supported. To do a similar function, see the "RESTLIBR Procedure" on page 4-203 or use the Restore Library (RSTLIB) command.

See the *CL Reference* book for more information on the RSTLIB command. See the *Backup and Recovery – Advanced* book for more information on how to save and restore your system libraries.

---

## REMOVE Procedure

The REMOVE procedure removes one or more specified library members from a library. You cannot use this procedure to remove library members from the following system libraries:

- QSYS

- QSSP
- QSYS38

To remove files or entire libraries from disk or diskette, see the “DELETE Procedure” on page 4-76.

The REMOVE procedure runs the \$MAINT utility program.

```
REMOVE { member name
        { member name ,ALL
        ALL
        }, [ SOURCE
            ( S )
            PROC
            ( P )
            LOAD
            ( O )
            SUBR
            ( R )
            LIBRARY
            ] , [ library name
                current library
            ]
```

59020194-0

### member name

This specifies the library members to be removed.

### member name,ALL

This specifies the beginning characters of the names of one or more library members to be removed. Specify up to 7 characters.

**ALL** This specifies that all library members are removed. When you specify ALL and LIBRARY, no other operators can use that library.

### SOURCE or S

This specifies removal of only library source members. If you do not specify a parameter, SOURCE is assumed. Source members are all the members in a source file named QS36SRC in the specified library.

### PROC or P

This specifies removal of only library procedure members. Procedure members are all the members in a source file named QS36PRC in the specified library.

### LOAD or O

This specifies removal of only library load members. Load members are all objects in a library with the following OS/400 object types:

<b>*PGM</b>	Programs
<b>*MSGF</b>	Message files
<b>*FILE</b>	Display files (attribute of *DSPF)

### SUBR or R

This specifies removal of only library subroutine members. Subroutine members are all objects in a library with the OS/400 object type of \*PGM (programs).

**LIBRARY** This specifies removal of all library types (SOURCE, PROC, LOAD, and SUBR), including all objects with the following OS/400 object types:

<b>*CHTFMT</b>	Chart formats
<b>*CLS</b>	Classes
<b>*CMD</b>	Commands
<b>*DTAARA</b>	Data areas
<b>*FCT</b>	Forms control tables
<b>*FILE</b>	Files (files with all attributes: PF, LF, PRTF, and so on)
<b>*GSS</b>	Graphic symbol sets

# RENAME

*JOB	Job descriptions
*JOBQ	Job queues
*MENU	Menus
*MSGQ	Message queues
*OUTQ	Output queues
*QRYDFN	Query definitions
*SBSD	Subsystem descriptions
*SSND	Session descriptions
*TBL	Tables

## Example 1

The following example removes the procedure member named PAYROLL from the library called MYLIB:

```
REMOVE PAYROLL,PROC,MYLIB
```

## Example 2

The following example removes all library members beginning with the characters PAY from the library named YOURLIB:

```
REMOVE PAY,ALL,LIBRARY,YOURLIB
```

## Example 3

The following example removes a source member named THIS, a procedure member named THAT, and all source members beginning with THEM from the library MYLIB (the current library):

```
REMOVE THIS  
REMOVE THAT,P  
REMOVE THEM,ALL
```

---

## RENAME Procedure

The RENAME procedure changes the name of an existing disk file, library, or folder. You can only rename a disk file, library, or folder if it is not currently in use. You can use the RENAME procedure to rename files on a remote system by specifying a current name that references a distributed data management (DDM) file. If you use RENAME to rename a remote file, the new file label must be defined as a DDM file as well. The two DDM files must indicate that the files are at the same remote location. The system libraries (#LIBRARY, QSYS, QSSP, QTEMP), job files, System/36 environment files library, and scratch files cannot be renamed.

You must be enrolled in the system distribution directory to rename folders.

The RENAME procedure runs the \$RENAM utility program.

```
RENAME current name,new name, [ mddy  
                               ddmmyy  
                               yymdd ]
```

59020195-0

### current name

This specifies the current name of the file, library, or folder.

If more than one file exists with the specified name, a message is

issued. You may choose to rename all of the files with the specified name or to cancel the job.

#### **new name**

This specifies the new name for the file, library, or folder. Specify a name of up to 8 characters in length. Begin the new name with an alphabetic character (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). Avoid commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), equal signs (=), and hyphens (-) because these characters have special meanings in procedures.

The new name cannot be the name of another disk file, library, or folder, even if that file, library, or folder has a different creation date. Do not use ALL as a file name; ALL, DISK, F1, #LIBRARY, PRINT, TAPE, or READER as a library name; or ALL or #LIBRARY as a folder name.

#### **mmddy, ddmmy, or yymmdd**

This specifies the creation date of the file (libraries and folders do not have a date). The date, if specified, must be in the session date format. Use the STATUS SESSION command to determine the session date format. The RENAME procedure does not change the creation date. The date is ignored for libraries and folders.

#### **Example 1**

The following example changes the name of an existing file from OLDPAY to NEWPAY:

```
RENAME OLDPAY,NEWPAY
```

#### **Example 2**

The following example changes the name of an existing file from OLDPAY to NEWPAY, changes THIS to THAT, and changes MYLIB to YOURLIB:

```
RENAME OLDPAY,NEWPAY
RENAME THIS,THAT
RENAME MYLIB,YOURLIB
```

**Note:** Although the System/36 environment does not allow you to create a file and a library with the same name, the AS/400 commands will not prevent duplicate names. Therefore, it is possible to have a file and a library with the same name. RENAME renames a file if it finds one with the specified name. If there is no file, it will rename a library. If there is no file or library, it will rename a folder.

---

## **REQUESTX Procedure**

Not supported. On System/36, the REQUESTX procedure allowed you to request or cancel an available user facility in an X.21 public data network. Depending on the network, registration for a variety of facilities or services was done by service order at subscription time, or over the network itself by using the REQUESTX procedure.

On the AS/400 system, registration for X.21 facilities or services is done by service order at subscription time.

---

### RESPONSE Procedure

The RESPONSE procedure updates the automatic response values and the severity levels for messages in a message file. Both IBM-supplied messages and user-written messages can be assigned these values.

Many IBM-supplied messages are shipped with default severity levels. Many of these messages have automatic responses which are called **default reply values**. To see the shipped values use the Display Message Description command (DSPMSGD) described in the *CL Programming* book.

The meaning of severity and response are different for the OS/400 program and the System/36 environment. In the System/36 environment, the severity is used with the NOHALT severity level to determine if the default reply should be used to respond to the message. The OS/400 program does not use severity with program messages to determine whether to apply the default reply. For these messages, the user specifies that the messages are to be the default reply (either on a job-basis for job-message-queue-related messages or on a message-queue-basis for messages sent to user- or device-message-queues).

The meaning of severity and response are also different in that OS/400 batch jobs can be set up for automatic cancellation if the cancel severity is less than or equal to the severity of a message being sent to the job message queue. This concept of cancel severity does not apply to System/36 environment messages.

You must run the NOHALT procedure or OCL statement in order for the RESPONSE procedure to be effective. The System/36 environment system-, session-, or job-severity-level at which the system automatically responds is set by the NOHALT procedure or OCL statement. See the "NOHALT Procedure" on page 4-161 and the "NOHALT OCL Statement" on page 5-65 for more information about setting the severity level.

The input to the RESPONSE procedure is an automatic response source member that contains the following (for each message to be responded to):

- Alpha code
- Message identification code (MIC)
- Response
- Severity level

The source member can be created using the source entry utility (SEU) or the \$MAINT utility program.

The changes the RESPONSE procedure make remain in effect until changed by another RESPONSE procedure or until a new release is installed for IBM-supplied messages. Changes made to a user message member remain in effect until changed by another RESPONSE procedure or until the message or message file is replaced. See "NOHALT Procedure" on page 4-161 for information about how long the NOHALT procedure remains in effect. See "Automatic Response Programming Considerations" on page 4-201 for other considerations.

The automatic response source member contains three types of statements:

- The automatic response control statement
- One or more automatic response specification statements
- One or more comment statements (optional)

See “Automatic Response Source Statements” on page 4-197 for information about the format of these statements.

The RESPONSE procedure runs the \$ARSP utility program.

RESPONSE source member name, <table border="1"> <tr> <td>library name</td> </tr> <tr> <td>current library</td> </tr> </table>	library name	current library
library name		
current library		

S9020198-0

#### source member name

This specifies the source member that contains the automatic response control statements, specification statements, and comment statements. The source member must be in a file named QS36SRC.

#### library name

This specifies the library that contains the source member. If you do not specify a library name, the current library is assumed.

#### Example

The following example applies the automatic responses contained in the library source member named AUTORESP (which is stored in a library named MYLIB):

```
RESPONSE AUTORESP,MYLIB
```

## Automatic Response Source Statements

This section explains automatic response source statements.

### The Automatic Response Control Statement

The automatic response control statement specifies the alpha code for the automatic response specification statements that follow. The alpha code identifies the IBM product or user message file in which the messages are to be changed. With the exception of the alpha code USER, other alpha codes define IBM-supplied message files.

Alpha codes also relate to the message identifier that is displayed when a message is output from the System/36 environment. Message identifiers on the OS/400 program consist of a three alphabetic-character prefix followed by a 4-digit number. For System/36 environment messages, the number is the same as it was on the System/36. The prefix for IBM-supplied messages is normally the first three characters of the alpha code. Two exceptions do exist:

- USER is changed to USR
- EP is changed to EPX

Additionally, when DBCS messages exist in the same file as single-byte messages, the third character of the prefix for the double-byte messages is always a Z. For example, if message USR2721 appears, USER is the alpha code and 2721 is the MIC number. A DBCS version of this message would be USZ2721.

The automatic response control statement must precede associated automatic response specification statements. A source member may contain more than one automatic response control statement and each can have its own automatic response specification statement and comment statements.

## RESPONSE

The RESPONSE procedure processes both non-DBCS and DBCS data in the case where a message may exist in both non-DBCS and DBCS form.

The syntax of the automatic response control statement is as follows:

```
alpha code, [load member name], [library name] [comment]
```

59020199-0

### alpha code

This specifies the alpha code of the messages being given automatic response values. The following alpha codes are supported in the System/36 environment:

BGU	DSU	IWS	RPG	SRTX
CBL	EMU	OFC	SDA	SYS
DFU	ESU	QRY	SEU	
DHCF	IDDU	RJE	SORT	

The following alpha codes were supported on the System/36, but are not supported in the System/36 environment. If one of these alpha codes are specified on an Automatic Response Control Statement, a message will be issued informing the user the message file for the specified alpha code was not found.

ASM	CGU	FOR	NRD	TXT
BAS	EP	KBD	TTM	WSU

Group together the messages with the same alpha codes in the source member. This grouping is done so the update to the source member can be made more efficiently.

### message file name (System/36 load member name)

This specifies the message file whose automatic response values are to be updated. If the alpha code is USER, specify the member name. If the alpha code is not USER, the member name parameter is ignored and the \$ARSP utility determines the message file name using the following search:

- The library list is searched for the message file.
- If the message file is not found in the library list, the product default library is searched. The product default library is one of the libraries in which the product was shipped.

Only the first message file found using this search is updated. The reason for searching the library list first is to update either your version of the message file or, when multilingual support exists, to update the version of the message file in your chosen language.

### library name

This specifies the library that contains the message file. If the alpha code is USER, the library name can either be specified or not specified. If you do not specify the library name, the system library (#LIBRARY) is assumed. If the alpha code is not USER, the library name parameter is ignored and the \$ARSP utility determines the library name.

**comment** This specifies any information that may help identify the message or the response you want. The system does not use this information. Place one or more blanks before the comment.

### Automatic Response Specification Statement

The automatic response specification statement specifies the message identification code (MIC) of the message to be updated and the automatic response and severity level values for that message.

If you specify an automatic response specification statement for a message that is not in the specified message member, the error message SYS8903 appears. To continue requires a user response. It is suggested that when you initially migrate your System/36 auto response members, you set up an auto response for this message. Doing this avoids having to respond to messages that were on the System/36, but do not exist in the System/36 environment.

To determine which messages are no longer supported, run RESPONSE and then display the detailed messages in your job log. Use the Display Job Log (DSPJOBLOG) command to view your job log.

In some cases where messages are no longer supported, the OS/400 program sends a similar message to the system operator. If you wish to automatically respond to an OS/400 message you should use automatic reply lists as described in the *CL Programming* book.

The format of the automatic response specification statement is as follows:

```
mic  response,severity level  [ comment ]
```

59020200-0

**MIC** This specifies the message identification code (MIC) of the message. The MIC must be a 4-digit number from 0000 to 9999, placed in positions 1 to 4 of the message text statement. The MICs must be in *ascending numeric* order.

**response** This specifies the automatic response value for the message. The allowed values are 0, 1, 2, 3, D, F, or N. Specifying a D results in a system dump and can be used on any message that allows a 3 option. Certain messages support an F option and will take a formatted dump of the program being executed before the job is canceled. If you specify N, the IBM-supplied automatic response value is used for the specified MIC. Place the automatic response in position 6 in the statement.

When a 3 option automatically cancels a job, a message appears, indicating the job was canceled by the system.

### severity level

This specifies the severity level for the message. The allowed values are 1, 2, 3, 4, and 5. The value must be placed in position 8 in the statement. If no value is specified, the current value will be unchanged.

You can specify that only messages with a specific severity level are to be responded to automatically. See "NOHALT Procedure" on page 4-161 and "NOHALT OCL Statement" on page 5-65 for informa-

## RESPONSE

tion on specifying this severity level for the system, your session, or a job.

The severity-level guidelines used for the IBM-supplied messages are as follows:

- 1 Informational messages (option 0 only).
- 2 Messages with one option, or messages with two options where one option is a retry option.
- 3 Program error messages; these usually have more than one option.
- 4 Messages for severe errors, such as hardware errors or permanent input/output errors.
- 5 No automatic response value is allowed for this message.

You can view both your message severity and automatic response using the Display Message Description (DSPMSGD) command. These values can also be modified using the Change Message Description (CHGMSGD) command. New messages can be added to the message file using the Add Message Description (ADDMSGD) command. When you use the ADDMSGD command you can define the severity and default reply as part of the message definition. Using these commands eliminates the need to run the RESPONSE procedure.

To use these commands you need to understand that the severity stored in the message description is not the same as the severity you enter on the response procedure. The OS/400 values range from 00 through 99. The values from 0 through 5 on the response procedure are changed before the message description is updated. To determine the corresponding OS/400 value, the values from 0 through 4 are multiplied by 10, and 5 is changed to a 99. During execution, the System/36 environment converts this number back to a value from 0 through 5 to determine if the message is to be responded to automatically.

For more information, see the message handling chapter in the *System/36 Environment Programming* book.

**comment** This specifies any information that helps identify the message or the response you want. The system does not use this information. Place one or more blanks before the comment.

### Comment (\*) Statement

This specifies any information that helps identify the message or the response you want. The system does not use this information. The format of the comment statement is as follows:

```
* comment
```

59020201-0

The asterisk (\*) must be the first character in the statement. Place comment statements anywhere within the other statements.

## Automatic Response Programming Considerations

Do not specify automatic response values to messages that:

- Indicate a dump has been taken
- Require an operator to do something before responding to the message
- Indicate a retry option

Choose values that do not result in lost data or will cause the system to loop. An example of a loop would be automatically responding to an error with a retry response that results in the same error. Consider the following when you are assigning values to messages:

- Messages get automatic responses only if the following is true:
  - The message requires a response.
  - The automatic response value is a value that the message allows.
- You may want to maintain two source members, one containing your automatic response values, the other containing the response value N (in column 6 of all automatic response specification statements).

By maintaining two source members, you can use one for your automatic response values and the other for the IBM-supplied automatic response values. To reset the IBM-supplied automatic response values, run the RESPONSE procedure using the source member with N in column 6.

### Example Automatic Response Source Member

The following example assumes that a source member contains the following statements. (Note how the statements are grouped by the alpha codes and the MICs are listed in numeric order.)

```
* Automatic responses for OS/400 displayed messages
SYS
1051 3,3 Invalid LPI parameter
1063 3,3 Invalid library name
1272 3,3 File statement cannot have both RETAIN-J and JOB-YES
*
* Automatic responses for RPG displayed messages
RPG
9011 0,3 Square root of negative number
9013 0,3 Divide by zero
9016 0,3 No data found
9037 0,3 Index key not in sequence
*
* Automatic responses for USER displayed messages
* The message load member is named MESSAGES,
* and is contained in the library MYLIB
USER,MESSAGES,MYLIB
0001 2,3 Error in customer number
0012 0,2 Error in item number
```

You could use the following source member to return the responses back to the IBM-supplied responses:

```
* Automatic responses for OS/400 displayed messages
SYS
1051 N Invalid LPI parameter
1063 N Invalid library name
1272 N File statement cannot have both RETAIN-J and JOB-YES
*
* Automatic responses for RPG displayed messages
RPG
9011 N Square root of negative number
9013 N Divide by zero
9016 N No data found
9037 N Index key not in sequence
```

---

## RESTEXTN Procedure

Not supported. On System/36, the RESTEXTN procedure restored from diskette all or part of the extended character file. The RESTEXTN procedure was only supported for the double-byte character set (DBCS) version of the SSP.

To do a similar function, use the Copy DBCS Font Table (CPYIGCTBL) command. The CPYIGCTBL command is only supported for the DBCS version of the OS/400 program.

See the *CL Reference* book for more information on the CPYIGCTBL command. See the *ADTS/400: Character Generator Utility* book for more information on DBCS font tables.

---

## RESTFLDR Procedure

Not supported. On System/36, the RESTFLDR procedure restored a folder copied onto disk, diskette, tape, or tape cartridge by the SAVEFLDR procedure.

To do a similar function, use the Restore Document Library Object (RSTDLO) command with the DLO (name) parameter specified.

See the *CL Reference* book for more information on the RSTDLO command. See the *System/36 Environment Programming* book for more information on using folders in the OS/400 System/36 environment. See the *Planning for and Setting Up OfficeVision/400* book for more information on folder management on the AS/400 system.

---

## RESTLIBR Procedure

The RESTLIBR procedure restores a library copied onto diskette, tape, or tape cartridge by the SVELIBR procedure. You can specify a new library name different from the one on the diskette, tape, or tape cartridge.

If the library name that is to be restored does not exist on the system, a new library is created in the system storage pool. If the library name that is to be restored exists on the system and you select to replace the library, the new library is located in the same system storage pool as the existing library.

To copy individual members to a library from a disk, diskette, or tape file, see "TOLIBR Procedure" on page 4-283. (Files copied by TOLIBR must be created either by the FROMLIBR procedure or by the \$MAINT utility program.) To restore data files, see "RESTORE Procedure" on page 4-206.

The RESTLIBR procedure runs the \$MAINT utility program, as follows:

```

RESTLIBR library name, [library size], [directory size], [
  A1
  A2
  A3
  A4
  block number
],
[
  S1
  S2
  S3
  M1.nn
  M2.nn
], [
  AUTO
  NOAUTO
], [
  I1
  T1
  T2
  TC
], [
  REWIND
  LEAVE
  UNLOAD
], [
  mddy
  ddmm
  yy
  yymm
  dd
], [new library name]

```

59020203-2

### library name

This specifies the library that is restored. You must specify a library name. You cannot restore to library QSYS.

If the library you are specifying already exists, a message appears that allows you to do the following:

- Cancel the RESTLIBR procedure.
- Delete the library from the disk and proceed with the RESTLIBR procedure.
- Replace the library on disk.

**library size**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**directory size**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**A1, A2, A3, or A4**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**block number**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**S1, S2, S3, M1.nn, or M2.nn**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**AUTO**

If specified for tape (unit is T1 or T2) when the tape reel on the original tape drive finishes, the system switches to the other tape drive to continue processing. If the tape reel on the other tape drive is not the same density as the original tape reel, an error message will be issued. If the other tape drive is not available, the system uses the original tape drive.

**NOAUTO**

If specified for tape (unit is T1, T2, or TC) when the tape reel on the original tape drive finishes, the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is tape and neither AUTO nor NOAUTO was specified, AUTO is assumed.

If TC is specified for the unit, the AUTO/NOAUTO parameter is ignored.

If the unit is diskette (I1), the AUTO/NOAUTO parameter is supported for compatibility only. If specified, the value will be syntax-checked, but will be ignored when the diskette is processed.

**I1**

This specifies that the library is to be restored from diskette. If you do not specify a parameter, I1 is assumed.

**T1, T2, or TC**

This specifies that the library is restored from tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

**REWIND**

This specifies, for reel-to-reel tape, that the tape rewinds to the load point after processing completes.

For a tape cartridge, REWIND specifies that the tape is positioned to the beginning after processing completes.

**LEAVE**

This specifies, for reel-to-reel and tape cartridges, that the tape is left where it is (not rewound) after processing completes. The next operation to the tape begins at this position.

**UNLOAD** This specifies, for a reel-to-reel tape, that the tape rewinds and unloads after the processing completes.

For a tape cartridge, UNLOAD specifies that the tape is positioned to the end of the cartridge after processing completes.

Each time a new or different tape cartridge processes or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, the operation is lengthy. The preparation time can be shortened if you specify UNLOAD, because the tape is left at the end of the tape after the last operation completes.

**mmddy, ddmmy, or yymmdd**

This specifies the creation date of the file to restore. Specify the date in the same format as the creation date of the diskette or tape file.

**new library name**

This specifies the name of the library to restore if you change the library name from the one specified on the diskette, tape, or tape cartridge. Specify a library name of up to 8 characters in length and begin the name with an alphabetic character (A–Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). Avoid using commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), minus signs (–), and equal signs (=) because these characters have special meanings in procedures. Do not use #LIBRARY, F1, READER, PRINT, DISK, TAPE, or ALL as a library name.

**Example 1**

The following example restores the system library:

```
RESTLIBR #LIBRARY
```

**Example 2**

The following example restores a library named PAYLIB.

```
RESTLIBR PAYLIB
```

**Example 3**

The following example restores a library named PAYLIB. The library is stored on a tape reel mounted on tape drive 1. After the library is restored, the tape rewinds and unloads.

```
RESTLIBR PAYLIB,,,A2,,,T1,UNLOAD
```

---

## RESTNRD Procedure

Not supported. On System/36, the RESTNRD procedure restored a version of the network resource directory (NRD) from diskette, tape, or tape cartridge to disk.

On the AS/400 system, the information that was stored in the NRD is kept in the OS/400 distributed data management (DDM) files. Each DDM file is the equivalent of one NRD entry.

## RESTORE

To do a similar function, use the Restore Object (RSTOBJ) command to restore your DDM files.

See the *CL Reference* book for more information on the RSTOBJ command. See the *Distributed Data Management* book for more information on using DDM files.

---

### RESTORE Procedure

The RESTORE procedure restores a saved file, a set of saved files, or part of a set of saved files from diskette, tape, or tape cartridge to disk. One of the following must have created the diskette, tape, or tape cartridge file, or files:

- The SAVE procedure
- The \$COPY utility program

When you restore only one file, you can change the size of the file by specifying the RECORDS or BLOCKS parameters.

The RESTORE procedure can also be used with physical, database data files that do not have a System/36 environment direct, indexed, or sequential file organization. When such files are used, the restored file may be created with file attributes that differ from those you requested or expected. A message allowing you to continue is sent whenever this may occur. See the *System/36 Environment Programming*, for more information on using System/36 environment files and AS/400 files.

You can use the RESTORE procedure to restore local diskette, tape, or tape cartridge files individually to a remote system; however, the RESTORE procedure cannot create a remote file which is externally described.

The RESTORE procedure runs the \$COPY utility program. The \$COPY utility will process diskette, tape, or tape cartridge files that were created on System/32, System/34, or System/36. You can also use the Restore System/36 File (RSTS36F) command to restore files saved on these systems.

**Note:** Libraries, folders, and system files not created by the \$COPY utility are not processed by the system.

See "RESTLIBR Procedure" on page 4-203 for information on restoring libraries. See "TRANSFER Procedure" on page 4-289 for information on copying basic data or I-exchange diskette files to the disk. See "TAPECOPY Procedure" on page 4-255 for information on copying exchange tape files to the disk.

Concurrent SAVE and RESTORE operations which use the same file library cannot save or restore files simultaneously. Instead, these operations alternate in saving or restoring files.

When you copy all files within a set from diskette, tape, or tape cartridge back to disk, the data begins with the first file within the set on the first diskette, tape reel, or tape cartridge, unless you specify a different starting file. On diskette, the first file within the set is also the first file on the diskette, because a set of files can only be saved on a diskette containing no unexpired files. Multiple SAVE ALL operations and RESTORE ALL operations which specify or default to the same set name cannot be run at the same time.

When you restore files from tape, the AUTO/NOAUTO parameter is ignored and only one tape drive may be used.

When you restore all the files in a set created on a system which allowed date-differentiated files to a system which does not allow date-differentiated files, date-differentiated files are restored if they exist in the set.

For restoring all previously saved data files of a set:

```
RESTORE [ ALL ] , [ set name ] , [ S1 ] , [ AUTO ] , [ REWIND ] , [ starting file name ] ,
        [ #SAVE ] , [ S2 ] , [ NOAUTO ] , [ LEAVE ] ,
        [ S3 ] , [ M1.nn ] , [ UNLOAD ] ,
        [ M2.nn ] ,
        [ T1 ] ,
        [ T2 ] ,
        [ TC ]

[ starting file date ]
```

59020205-1

For restoring a single previously saved data file:

```
RESTORE file name , [ mmddy ] , { RECORDS , records } { LOCATION , location } ,
        [ ddmmy ] { BLOCKS , blocks } { , location } ,
        [ yymmdd ]

[ S1 ] , [ AUTO ] , [ INCLUDE ] , [ position ] , [ EQ ] , [ 'characters' ] ,
[ S2 ] , [ NOAUTO ] , [ OMIT ] , [ NE ] ,
[ S3 ] , [ M1.nn ] , [ M2.nn ] , [ LT ] ,
[ T1 ] , [ T2 ] , [ GE ] ,
[ TC ]

[ record length ] , [ SAME ] , [ key position , key length ] , [ DUPKEY ] ,
[ S ] , [ NODUPKEY ] ,
[ I ] ,
[ D ]

[ REWIND ] ,
[ LEAVE ] ,
[ UNLOAD ]
```

59020206-1

**ALL** This specifies that all files previously saved in a set are to be restored to the disk. If you do not specify the first parameter, ALL is assumed.

**set name** This specifies the name used for the entire set of files saved on diskette, tape, or tape cartridge by the SAVE (SAVE ALL) procedure. If you do not specify a set name, a set name of #SAVE is assumed.

**file name** This specifies a single diskette, tape, or tape cartridge file that is to be restored to the disk. If more than one file exists with the specified name and if you do not specify the creation date, the system restores the first file it finds with the name you specified.

#### **mmddy, ddmmy, or yymmdd**

This specifies the date that the disk file, which is saved within the diskette, tape, or tape cartridge file of the same name, was originally created on disk. If the diskette, tape, or tape cartridge file was created through a SAVE ALL operation on a system which allowed date-differentiated files, the diskette, tape, or tape cartridge file may contain more than one disk file.

## RESTORE

To restore a specific disk file other than the file with the latest date, you must specify when the file to be restored was originally created on disk.

The date must be in the same format as the session date. Use the STATUS SESSION control command to determine the date format.

**Note:** This parameter does not refer to the actual creation date of the diskette, tape, or tape cartridge file.

If you previously saved more than one file or set with the same name on a tape, you may have to specify a sequence number to restore a file or set after its first occurrence on the tape. To specify a sequence number, load and run the \$COPY utility program. On the COPYIN FILE statement, specify the sequence number of the file you want to restore, or specify the sequence number of the first file or the starting file in the set that you want to restore.

### RECORDS or BLOCKS

This specifies the size for the file. If you do not specify BLOCKS or RECORDS and the associated size, the file is restored to its original size. If you change the organization and/or record length of the file, the restored file will be made large enough to hold the total number of records in the diskette, tape, or tape cartridge file.

**Note:** When you attempt to restore a single diskette file that has been added to using the SAVE(SAVE ADD) procedure or the \$COPY utility program, the restore may fail if you do not specify a larger size for the disk file. This failure may occur if your additions to the diskette file have made it larger than its original disk allocation size.

### RECORDS

This specifies that the disk file is to be made large enough to contain the number of records specified. **Records** can be any number from 1 to 8000000.

**BLOCKS** This specifies that the disk file is made large enough to contain the number of blocks specified. **Blocks** can be any number from 1 through the maximum number of blocks of disk storage configured on the system.

### LOCATION

**location** can be:

#### A1, A2, A3, or A4

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

#### block number

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

#### S1, S2, S3, M1.nn, or M2.nn

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

#### T1, T2 or TC

These specify the tape drive containing the first tape to be processed. **T1** indicates the first tape drive, **T2** indicates the second tape drive, and **TC** indicates the tape cartridge drive.

**AUTO** The AUTO/NOAUTO parameter is supported for compatibility only. If you specify AUTO or NOAUTO, the value is syntax-checked, but ignored when the diskette or tape is processed. Only one tape drive may be used during a RESTORE operation.

**NOAUTO** The AUTO/NOAUTO parameter is supported for compatibility only. If you specify AUTO or NOAUTO, the value is syntax-checked, but ignored when the diskette or tape is processed. Only one tape drive may be used during a RESTORE operation.

**INCLUDE or OMIT**

This specifies whether specific records in the file to be restored are to be included in or omitted from the restored file. The INCLUDE and OMIT parameters work with the position, EQ, NE, LT, GT, LE, GE, and 'characters' parameters. If you specify only 'characters' or position, INCLUDE and EQ are assumed. You cannot use INCLUDE, OMIT, position, EQ, NE, LT, LE, GT, GE, or 'characters' during a direct-to-direct file copy operation because the use of these parameters can result in the loss of record positioning in the direct output file.

**position** This specifies, for each record, the first character to compare with the comparison characters. The position can be any number from 1 to 4096. If you do not specify a position and INCLUDE and EQ have been specified (or may be assumed), every position in the record is compared with the comparison characters until the specified condition is met.

**EQ** This specifies that if the characters in the record indicated by position are the same as the comparison characters, the record is to be included in or omitted from the restored file.

**NE** This specifies that if the characters in the record indicated by position are not the same as the comparison characters, the record is to be included in or omitted from the restored file.

**LT** This specifies that if the characters in the record indicated by position are less than the comparison characters, the record is to be included in or omitted from the restored file.

**LE** This specifies that if the characters in the record indicated by position are less than or the same as the comparison characters, the record is to be included in or omitted from the restored file.

**GT** This specifies that if the characters in the record indicated by position are greater than the comparison characters, the record is to be included in or omitted from the restored file.

**GE** This specifies that if the characters in the record indicated by position are greater than or the same as the comparison characters, the record is to be included in or omitted from the restored file.

**'characters'**

This specifies the comparison characters. You can specify up to 30 characters. The characters should be enclosed by apostrophes ('). You can include blanks and commas (,), but you cannot include apostrophes as data.

**record length**

This specifies the record length of the restored file. This can be any number from 1 to 4096. If you do not enter this parameter, the record

## RESTORE

length of the file on diskette, tape, or tape cartridge is used for the record length of the restored file.

If the record length of the file on diskette, tape, or tape cartridge is less than the specified record length, the additional record positions in the restored file are filled with blanks. If the record length of the file on diskette, tape, or tape cartridge is greater than the specified record length, the extra positions in the diskette, tape, or tape cartridge file are truncated. If the file to be restored is an indexed file and the key field would be truncated, an error message is displayed.

- SAME** This specifies that the restored file is to have the same organization as the file that was saved. If you do not specify a parameter, SAME is assumed.
- S** This specifies that the restored file is to be organized as a sequential file.
- I** This specifies that the restored file is to be organized as an indexed file.
- D** This specifies that the restored file is to be organized as a direct file.

### key position

This specifies the starting position of the key for the restored file. Specify the key position if the restored file is to be changed into an indexed file from a sequential or direct file (that is, you specify I). If you are restoring an indexed file, you can specify another field for the key. The key position can be any number from 1 to 4096. The entire key, defined by the key position and key length, must be within the record.

If you do not specify a value, and the file you are restoring is an indexed file, the key position of the saved indexed file is assumed. If you specify a key position, you must also specify the key length.

### key length

This specifies the length of the key for the restored file. Specify the key length if the restored file is to be changed from a sequential or direct file into an indexed file (that is, you specified I). If you are restoring an indexed file, you can specify another field for the key. The key length can be any number from 1 to 120. The entire key, defined by the key position and key length, must be within the record.

If you do not specify a value, and the file you are restoring is an indexed file, the key length of the saved indexed file is assumed. If you specify a key length, you must also specify the key position.

- DUPKEY** This specifies that duplicate keys are to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If you do not specify this parameter, the attribute of the saved file will become the attribute of the restored indexed file.

### NODUPKEY

This specifies that duplicate keys are not to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If you do not specify this parameter, the attribute of the saved file will become the attribute of the restored indexed file.

**Note:** NODUPKEY does not remove duplicate key records.

**REWIND** This specifies, for reel-to-reel tape, that the tape rewinds to the load point after processing is completed.

For a tape cartridge, REWIND specifies that the tape rewinds to the beginning of the cartridge after processing is completed.

**LEAVE** This specifies, for reel-to-reel tape or a tape cartridge, that the tape is left where it was last processed. The next operation to the tape begins at that point.

**UNLOAD** This specifies, for reel-to-reel tape, that the tape rewinds and is unloaded after processing is completed.

For a tape cartridge, UNLOAD specifies that the tape is positioned to the end of the cartridge after processing is completed.

REWIND, LEAVE, and UNLOAD are valid only if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge is processed or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, this process is lengthy. You can shorten the preparation time by specifying UNLOAD, because the tape is left at the end of the tape after the last operation completes.

#### **starting file name**

This specifies the diskette, tape, or tape cartridge file within a set of files with which the RESTORE ALL operation should begin. This file and all files located after it are restored. If you do not specify a parameter, the system begins restoring files with the first file of the set.

#### **starting file date**

If specified, is syntax-checked, but otherwise ignored. This parameter is supported for compatibility only. It was used to specify the creation date of the starting diskette, tape, or tape cartridge file specified by the starting file name. If specified, the date must be in the same format as the session date. Use the STATUS SESSION control command to determine the date format. If you specify a starting file date, you must also specify a starting file name.

#### **Example 1**

The following example restores a file named PAYROLL to disk from diskette:

```
RESTORE PAYROLL
```

#### **Example 2**

The following example restores part of a set of files saved on diskette using the SAVE ALL procedure. All of the files on diskette after and including the file named FILEA are restored.

```
RESTORE ALL,,,,,FILEA
```

### Example 3

The following example restores a file named PAYROLL to disk from tape. The tape is mounted on tape drive 1.

```
RESTORE PAYROLL,,,,,T1
```

### Example 4

The following example restores a file named PAYROLL from tape cartridge to disk:

```
RESTORE PAYROLL,,,,,TC
```

### Example 5

This example restores a file named FILE1. The key is to be changed from positions 1 through 4 to positions 5 through 24, and only those records that contain the phrase 'NEW' anywhere in the record are to be copied.

```
RESTORE FILE1,,,,,S1,NOAUTO,INCLUDE,,EQ,'NEW',,I,5,20
```

---

## RETRIEVE Procedure

Not supported. On System/36, the RETRIEVE procedure restored a folder member to a folder on disk, that was copied onto disk, diskette, tape, or tape cartridge by the ARCHIVE procedure. The archived member could be restored to the same folder or a different folder.

To do a similar function, use the Restore Document Library Object (RSTDLO) command.

See the *CL Reference* for more information on the RSTDLO command. See the *System/36 Environment Programming* for more information on using folders in the OS/400 System/36 environment. See the *Planning for and Setting Up OfficeVision/400* book for more information on folder management on the AS/400 system.

---

## RGZFILE Procedure

The RGZFILE procedure is supported only for compatibility with the IBM System/36. Only the procedure interface is supported.

- See the Reorganize Physical File Member (RGZPFM) command for information on removing deleted records and reordering records by key.
- See the Copy File (CPYF) command for information on including or omitting records copied between files and copying records by key.
- See the Change Physical File (CHGPF) command for information on changing the allocation size of a file.

Only physical disk files that do not have a direct file organization can be reorganized.

The third and fourth parameters cannot be used to change the allocation size of files with more than one member. A message allowing you to continue, without changing the allocation size of the file, is sent whenever this situation occurs.

The fifth parameter identifies, on the System/36, a new disk preference or a new block location for the file. Syntax checking is done, but the value is not used.

Every position in each record will be compared with the character string specified in the tenth parameter if both of the following conditions are true:

- You do not supply a position value for the eighth parameter
- INCLUDE and EQ processing are specified, or can be assumed for the seventh and ninth parameters

The eleventh parameter indicates, on the System/36, whether an indexed file should allow duplicate keys. Syntax checking is done, but the value is not used. If you attempt to change the duplicate keys attribute of a file, a message is sent. The message allows you to continue without changing the duplicate keys attribute.

When you use the RGZFILE procedure to perform INCLUDE or OMIT processing, a temporary intermediate disk file may be created.

The RGZFILE procedure does not change the creation date of the file member that is reorganized.

---

## RJFILE Procedure

The RJFILE procedure lets you convert a punch data file or a print file that the remote job entry facility (RJE) wrote to a compressed file, to a file that you define.

This procedure uses the Convert RJE Data (CVTRJEDTA) command.

```
RJFILE file name, [control table name]
```

59020209-0

**file name** This specifies the file used by the utility.

**control table name**

This specifies the control table where the entry is located that is used by the utility. This control table is created by the RJTABLE procedure. If you do not specify this parameter, the utility uses values supplied by the procedure. See the “RJTABLE Procedure” for more information.

### Example

The following example shows the RJFILE procedure processing a file named DATAFILE using a control table named CONTROLS:

```
RJFILE DATAFILE,CONTROLS
```

---

## RJTABLE Procedure

The RJTABLE procedure displays the Forms Control Table system menu. From this menu you can work with the forms control table or type commands.

```
RJTABLE
```

59020210-0

The RJTABLE procedure has no parameters.

**Example**

The following example starts the RJTABLE procedure:

```
RJTABLE
```

**ROLLKEYS Procedure**

Not supported. On System/36, the ROLLKEYS procedure allowed you to set the direction that the roll keys move information on your display.

To do a similar function, use the \*ROLLKEY value for the user options (USROPT) parameter on the Change Profile (CHGPRF) or Change User Profile (CHGUSRPRF) command.

See the *CL Reference* book for more information on the CHGPRF and CHGUSRPRF commands.

**RPG Procedure**

The RPG procedure is supported for System/34 compatibility only. See the "RPGC Procedure" information below on compiling RPG programs.

**RPGC Procedure**

The RPGC procedure compiles an RPG II program. For information on RPG II, see the *System/36-Compatible RPG II User's Guide and Reference* book.

```

RPGC  source member name, [ source member library ], [ NODSM ],
      [ current library ], [ DSM ],

      [ PRINT ], [ NOXREF ], [ mrt maximum ], [ NONEP ],
      [ NOPRINT ], [ XREF ], [ 0 ], [ NEP ],
      [ CRT ],

      [ output library ], [ SOURCE ], [ DEBUG ], [ program size ],
      [ source member library ], [ PSOURCE ], [ NODEBUG ],

      [ NOHALT ], [ REPLACE ], [ LINK ], [ NOOBJECT ], [ subroutine library ],
      [ HALT ], [ NOREPLAC ], [ NOLINK ], [ OBJECT ], [ source member library ],

      [ GEN ], [ work file size ], [ data dictionary name ], [ NOMRO ],
      [ NOGEN ], [ 40 ], [ MRO ]

```

59020211-1

**source member name**

This specifies the library source member that contains the RPG II program specifications.

**source member library**

This specifies the library that contains the source member to be compiled. If you do not specify this parameter, the current library is assumed.

**NODSM or DSM**

This is supported for System/36 compatibility only. The value is not used.

**PRINT** This specifies that the compiler listings created by the RPGC procedure print. If you do not specify a printer, PRINT is assumed.

**NOPRINT** This specifies that no compiler listings print or display.

**CRT** This specifies that the compiler listings created by the RPGC procedure display at the display station running the RPGC procedure.

**NOXREF** This specifies that the RPGC procedure does not produce a cross-reference listing of the RPG II program. If you do not enter a parameter, NOXREF is assumed.

**XREF** This specifies the production of a cross-reference listing.

**MRT maximum**

This specifies the maximum number of display stations that can use the program at any one time. Enter any number from 0 to 99. If you do not enter the parameter, a value of 0 is assumed. A value of 0 indicates that the program is a single requester terminal (SRT) program. Each display station that runs the program runs its own copy of the program. A value of 1 or more indicates that the program is a multiple requester terminal (MRT) program.

For more information on MRT programs, see the *System/36 Environment Programming* book.

**NONEP** This specifies that the program is not a never-ending program (NEP). If you do not enter a parameter, NONEP is assumed.

**NEP** This specifies that the program is a NEP. See the NEP parameter of "ATTR OCL Statement" on page 5-8 for more information on NEPs.

**output library**

This specifies the name of the library to contain the compiled program. If you do not specify this parameter, the source member library is assumed.

**SOURCE, PSOURCE, or NOSOURCE**

This is supported for System/36 compatibility only. The value is not used.

**DEBUG or NODEBUG**

This is supported for System/36 compatibility only. The value is not used.

**program size**

This is supported for System/36 compatibility only. The value is not used.

**NOHALT or HALT**

This is supported for System/36 compatibility only. The value is not used.

**REPLACE**

This specifies that if you are creating a program, and a load member or a subroutine member with the same name as your program already exists in the output library, the newly compiled program replaces the

existing load or subroutine member. No message appears indicating the replace. If you do not specify a parameter, REPLACE is assumed.

### **NOREPLAC**

This specifies that if you are creating a program and a load member or subroutine member with the same name as your program already exists in the output library, a message appears and you can either replace the member or cancel the procedure.

**LINK** This specifies that a program is to be created. If you do not enter a parameter, LINK is assumed.

**NOLINK** This specifies that no program is to be created.

### **NOOBJECT or OBJECT**

This is supported for System/36 compatibility only. The value is not used.

### **subroutine library**

This is supported for System/36 compatibility only. The value is not used.

**GEN** This specifies that if the RPG program being compiled contains a CONSOLE file, the display formats for that CONSOLE file are generated as part of the compile process. If you do not specify a parameter, GEN is assumed.

**NOGEN** This specifies that no display formats for a CONSOLE file are generated.

### **work file size**

This is supported for System/36 compatibility only. The value is not used.

### **data dictionary name**

This specifies the library that contains the communications file definition used with the program being compiled. On System/36, the communications file definition was stored in a data dictionary.

### **MRO or NOMRO**

This is supported for System/36 compatibility only. The value is not used.

### **Example**

The following example compiles an RPG II program named PAYROLL. The program is contained in the current library. The compiled program (load member) is placed in the current library. A source listing and a cross-reference are generated.

```
RPGC PAYROLL,,,XREF,,,SOURCE
```

---

## **RPGLOAD Procedure**

Not supported. On System/36, the RPGLOAD procedure installed the RPG support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## RPGONL Procedure

Not supported. On System/36, the RPGONL procedure allowed you to develop a RPG program. You could create a new program, or make a large number of changes to an existing program. The RPGONL procedure showed a series of displays that allow you to enter, compile, and change RPG programs.

Similar functions are provided by the System/36 Programming Languages Procedures (S36PGMLNG) menu. This menu provides options for running the AUTOC, RPGC, RPGR, and RPGSEU procedures. See the "AUTOC Procedure" on page 4-11, "RPGC Procedure" on page 4-214, "RPGR Procedure," and "RPGSEU Procedure" on page 4-219 for more information.

---

## RPGP Procedure

Not supported. On System/36, the RPGP procedure displayed a menu that allowed you to select the RPG II task you wanted to perform. You could enter, change, or compile an RPG II program. You could also request a cross-reference listing, or create or change display formats used with RPG II programs.

Similar functions are provided by the System/36 Programming Languages Procedures (S36PGMLNG) menu. This menu provides options for running the AUTOC, RPGC, RPGR, RPGSDA, RPGSEU, and RPGX procedures. See the "AUTOC Procedure" on page 4-11, "RPGC Procedure" on page 4-214, "RPGR Procedure," "RPGSDA Procedure" on page 4-218, "RPGSEU Procedure" on page 4-219, and "RPGX Procedure" on page 4-219 for more information.

---

## RPGR Procedure

The RPGR procedure generates display format source and load members for a CONSOLE file in an RPG II source program. For information on RPG II, see the *System/36-Compatible RPG II User's Guide and Reference* book..

```

RPGR  source member name, [work file size], [SAVE
                             40          NOSAVE ],
      [source member library], [load member library], [GEN],
      [REPLACE], [PRINT]
      [NOREPLAC], [NOPRINT]

```

59020215-0

### source member name

This specifies the library source member that contains the RPG II program specifications.

### work file size

This is supported for System/36 compatibility only. The value is not used.

### SAVE

Indicates that the source statements for the display format member are to be saved. If no parameter is entered, SAVE is assumed.

**NOSAVE** Indicates that the source statements for the display format member are not to be saved. Only the display format load member will be created.

**source member library**

This specifies the library that contains the RPG source member. If no library name is specified, the current library is assumed.

**load member library**

This specifies the library that is to contain the display format load member. If no library name is specified, the current library is assumed.

**GEN** This specifies that the display formats for the CONSOLE file are to be created. If no parameter is specified, GEN is assumed.

**REPLACE**

This specifies that an existing display format load member with the same name as **member name** is to be replaced. If no parameter is specified, REPLACE is assumed.

**NOREPLAC**

This specifies that if a load member already exists with the same name, a message is to be displayed and you can either continue or cancel the procedure.

**PRINT** This specifies that a listing of the created display formats is to be printed. If no parameter is specified, PRINT is assumed.

**NOPRINT** This specifies that the listing is not to be printed.

**Example**

This example shows how to generate the CONSOLE file display formats for an RPG II program named PAYROLL.

```
RPGR PAYROLL
```

---

## RPGSAVE Procedure

Not supported. On System/36, the RPGSAVE procedure saved the RPG support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## RPGSDA Procedure

The RPGSDA procedure starts the screen design aid (SDA) procedure. See the *ADTS/400: Screen Design Aid* book for more information about how to use SDA and for information about display formats.

See the *System/36-Compatible RPG II User's Guide and Reference* book for more information about this procedure and about RPG.



59020217-0

The RPGSDA procedure has no parameters.

## Example

The following example starts the RPGSDA procedure:

```
RPGSDA
```

---

## RPGSEU Procedure

The RPGSEU procedure allows you to create or change an RPG II program or procedure using the source entry utility (SEU). For information on RPG II, see the *System/36-Compatible RPG II User's Guide and Reference* book. For more information on SEU, see the *ADTS/400: Source Entry Utility* book.

```
RPGSEU member name, [ R
                     A
                     P ], [ seu format member ], [ statement length ],
                        [ library name
                          current library ]
```

59020218-0

### member name

This specifies the library member to create or change.

- R** This specifies an RPG II source member. If you do not specify a parameter, R is assumed.
- A** This specifies an RPG II source member containing auto report specifications.
- P** This specifies a procedure member.

### SEU format member

This is supported for System/36 compatibility only. The value is not used.

### statement length

This is supported for System/36 compatibility only. The value is not used.

### library name

This specifies the library that contains or will contain the member being changed or created. If you do not specify a library name, the current library is assumed.

## Example

The following example uses the RPGSEU procedure to change an RPG II source member named PAYROLL. The current library contains the source member.

```
RPGSEU PAYROLL
```

---

## RPGX Procedure

The RPGX procedure creates a cross-reference listing for an RPG II program (without compiling the program). For information on RPG II, see the *System/36-Compatible RPG II User's Guide and Reference* book.

## SAVE

```
RPGX  source member name, [work file size] , [source member library]
                               40                               current library
```

S9020219-0

If you enter the RPGX procedure with no parameters, a display appears and you enter the parameters.

### source member name

This specifies the source member that contains the RPG II program specifications.

### work file size

This is supported for System/36 compatibility only. The value is not used.

### source member library

This specifies the name of the library that contains the source member to be cross-referenced. If you do not specify this parameter, the current library is assumed.

### Example

The following example creates a cross-reference listing for an RPG II program named PAYROLL.

```
RPGX PAYROLL
```

---

## SAVE Procedure

The SAVE procedure allows you to do the following:

- Save a single disk file on diskette, tape, or tape cartridge. You can also use the SAVE procedure to select records to be saved (only when saving a single file with no ADD operations).
- Add another disk file to a disk file already saved on diskette.
- Save a specified file group on diskette, tape, or tape cartridge.
- Save all user disk files on diskette, tape, or tape cartridge (including file groups).
- Save all user disk files on diskette, tape, or tape cartridge (except file groups).
- Save files on diskette in a compressed format.
- Save multiple file sets on tape or tape cartridge.

The SAVE procedure saves only user disk data files. You cannot use the SAVE procedure to save a library or folder.

See "SAVELIBR Procedure" on page 4-228 for information on saving a library on diskette, tape, or tape cartridge.

To use the SAVE procedure to save remote files individually to local diskettes, tapes, or tape cartridges, you must request that the contents of the remote files be changed by the SAVE procedure in some way.

The AS/400 system requires that diskettes used in SAVE operations be initialized in the SAVRST format (1024-byte format). See the Initialize Diskette (INZDKT) command in the *CL Reference* book.

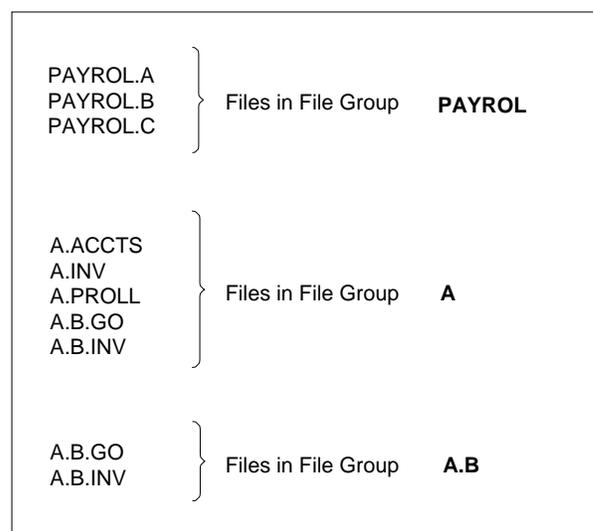
Files you save using the SAVE procedure cannot be restored on System/36, System/34, or System/32. Use the Save System/36 File (SAVS36F) command to save files that are to be restored on System/36.

When you use the SAVE procedure to save an indexed file, the system saves the data and a description of the index, but not the index itself.

When you save an alternative index, the system saves only a description of the index. Saving the physical file does not save the alternative indexes. However, you can save a physical file and all indexes for that file by using the file group-naming convention and specifying that group name for the file group name parameter on the SAVE procedure.

File groups are defined by file names that contain a period. The characters preceding the period identify the file group, and the characters following the period identify the file within the group. As for all file names, the maximum number of characters is 8, including the period. Files with names that do not contain a period are not part of a file group.

Figure 4-5 shows examples of names of files within a file group.



S9020222-0

Figure 4-5. Examples of Names of Files within a File Group

You may use the SAVE procedure to save a disk file being used by another job on the system if that job's FILE statement specifies DISP-SHRRR or DISP-SHRRM. For more information about file sharing, see "FILE OCL Statement (for Disk Files)" on page 5-25.

You may run the SAVE procedure from any display station. Concurrent SAVE operations and RESTORE operations that use the same file library cannot save or restore files simultaneously. Instead, these operations will alternate in saving or restoring files. Multiple SAVE ALL and RESTORE ALL operations which specify or have a default to the same set name cannot be run at the same time.

During a SAVE ALL operation, two files are locked at a time (the file being saved and the next file to be saved). Any files created during the SAVE ALL operation will not be saved.

# SAVE

When you copy to diskette, tape, or tape cartridge all members of a specified file group, all disk files including file groups, or all disk files except file groups, you create a set of files on the diskette, tape, or tape cartridge. When you create this set of files on diskette, the receiving diskettes must not contain any unexpired files. However, when you create this set of files on tape or tape cartridge, the first tape or tape cartridge may contain previously saved unexpired files or sets of files. Be cautious of duplicate file or set names.

## Notes:

1. You cannot write over expired tape files using the SAVE procedure. Initialize the tape first. You can copy to other tape reels or cartridges if the first file of each reel or cartridge has expired.
2. Be cautious of duplicate file and set names when saving a file or set to tape. If you create more than one file or set with the same name on a tape, you may have to specify a sequence number to restore any occurrences of that file or set after the first occurrence on the tape. To specify a sequence number, load and run the \$COPY utility program. On the COPYIN FILE statement, specify the sequence number of the file you want to restore, or specify the sequence number of the first file or the starting file in the set that you want to restore.

The SAVE procedure runs the \$COPY utility program. \$COPY does not process a library, a folder, or system files.

To save one disk file on diskette, tape, or tape cartridge:

```
SAVE    file name, [retention days], [mmddyymmddyymmdd], volume id, [S1  
S2  
S3  
M1.nn  
M2.nn  
T1  
T2  
TC], [AUTO  
NOAUTO],  
  
[NOREORG  
REORG], [INCLUDE  
OMIT], [position], [EQ  
NE  
LT  
GT  
LE  
GE], ['characters'],  
  
[REWIND  
LEAVE  
UNLOAD], [NOCOMPRESS  
COMPRESS]
```

59020220-1

To add a disk file to a diskette file:

```
SAVE    file name,ADD, [mmddyymmddyymmdd], volume id, [S1  
S2  
S3  
M1.nn  
M2.nn], [AUTO  
NOAUTO]
```

59020221-0

To save only disk files from a file group:

```
SAVE [ALL], [retention days], [set name], volume id, file group, [S1
S2
S3
M1.nn
M2.nn
T1
T2
TC],
[AUTO
NOAUTO], [REWIND
LEAVE
UNLOAD], [NOCOMPRESS
COMPRESS]
```

59020223-1

To save all disk files, including files that belong to a file group:

```
SAVE [ALL], [retention days], [set name], volume id, ALL, [S1
S2
S3
M1.nn
M2.nn
T1
T2
TC], [AUTO
NOAUTO],
[REWIND
LEAVE
UNLOAD], [NOCOMPRESS
COMPRESS]
```

59020224-1

To save all disk files except files that belong to a file group:

```
SAVE [ALL], [retention days], [set name], volume id,, [S1
S2
S3
M1.nn
M2.nn
T1
T2
TC], [AUTO
NOAUTO],
[REWIND
LEAVE
UNLOAD], [NOCOMPRESS
COMPRESS]
```

59020225-1

**file name** This specifies the disk file to be saved. The diskette, tape, or tape cartridge file name will be the same as the disk file name.

**Note:** When adding a disk file to a diskette file, both file names must be the same.

**ALL** This specifies the following:

- If you specify ALL as both the first and the fifth parameters, all disk files are saved, whether or not file groups exist.
- If you specify ALL as the first parameter and specify a file group as the fifth parameter, all members of that file group are saved.
- If you specify ALL as the first parameter and do not specify anything as the fifth parameter, all files that are not members of file groups are saved.

If you do not specify a first parameter, ALL is assumed.

**retention days**

This specifies the number of days the file should be retained. Any number between 0 and 999 may be specified, except when saving to

diskette all the disk files in a specific file group, all disk files including those in file groups, or all disk files except those in file groups. In these three cases, the number must be between 1 and 999. (If you specify 0 for any one of these three, the value is changed to 1.)

If you do not specify a retention period, 1 day is assumed. If you specify a retention period of 999 days, the diskette, tape, or tape cartridge file becomes a permanent file. For more information on diskette, tape, or tape cartridge file retention, see "FILE OCL Statement (for Diskette Files)" on page 5-33 or "FILE OCL Statement (for Tape Files)" on page 5-37.

**ADD** This specifies that a single disk file is to be added to a file previously saved on diskette. Specifying ADD places extension files on a diskette immediately following the file to which you are adding. The diskette file to which you are adding or an extension file of this diskette file must be the last file on the diskette. Use the CATALOG procedure (to list the files on a diskette by location) to determine whether the file you want to add to or an extension file of this file, is the last file on the diskette.

**Notes:**

1. The previously saved file and the file that is to be added to the previously saved file must have the same attributes.
2. If you specify ADD, you cannot specify parameters 7 through 13.
3. Specify ADD only for diskette, not for tape or tape cartridge.

**set name** This specifies the name associated with the entire set of saved files. If you do not specify a name, the set name #SAVE is used.

**mmddyy, ddmmyy, or yymmdd**

This specifies the creation date of the disk file. The date must be in the same format as the session date (use the STATUS SESSION control command to determine the date format). If you do not specify the creation date and more than one file with the specified name exists, the most recent file is saved.

**volume id**

This specifies the volume ID of the diskette, tape reel, or tape cartridge. Specify from 1 to 6 alphameric characters.

**file group**

This specifies the name of the file group to be saved. Do not specify the period (.), which indicates a file group name. For example, to save files belonging to the file group that includes PAYROL.A, PAYROL.B, and PAYROL.C, enter PAYROL for this parameter.

**S1, S2, S3, M1.nn and M2.nn**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**T1, T2, or TC**

These specify the tape drive containing the first tape to be processed. T1 indicates the first tape drive, T2 indicates the second tape drive, and TC indicates the tape cartridge.

**AUTO**

This specifies that when the tape reel (unit T1 or T2) on the original tape drive finishes, the system attempts to switch to the other tape drive to continue processing. If the tape reel on the other tape drive is not the

same density as the original tape reel, an error message is sent. If the other tape drive is not available, the system uses the original tape drive.

**NOAUTO** This specifies that when the original tape drive finishes (unit T1, T2, or TC), the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is tape, and neither AUTO nor NOAUTO is specified, AUTO is assumed.

If TC is specified for the unit, the AUTO/NOAUTO parameter is ignored.

If the unit is diskette (11), the AUTO/NOAUTO parameter is supported for compatibility only. If specified, the value is syntax-checked, but is ignored when the diskette is processed.

### **NOREORG**

This specifies that the file should not be reorganized. That is, for sequential and indexed files, any records that were deleted, are saved along with the records that contain data. For direct files, deleted records are always saved. Also, for indexed files, the records are saved in the order in which they occur in the file.

**REORG** This specifies reorganization of the file. That is, for sequential and indexed files, any records that were deleted are not saved. Only the records that contain data are saved. For direct files, deleted records are always saved. Also, for indexed files, the records are saved sequentially by key.

### **INCLUDE or OMIT**

This specifies whether specific records in the file should be included in or omitted from the save. The INCLUDE and OMIT parameters work with the position, EQ, NE, LT, GT, LE, GE, and 'characters' parameters. If you only specify the 'characters' or position, INCLUDE and EQ are assumed.

**position** This specifies, for each record, the first character to compare with the comparison characters. The position can be any number from 1 to 4096. If you do not specify a position, and INCLUDE and EQ have been specified (or may be assumed), every position in the record is compared with the comparison characters until the specified condition is met.

**EQ** This specifies that if the characters in the record indicated by position are the same as the comparison characters, the record is included in or omitted from the save.

**NE** This specifies that if the characters in the record indicated by position are not the same as the comparison characters, the record should be included in or omitted from the save.

**LT** This specifies that if the characters in the record indicated by position are less than the comparison characters, the record should be included in or omitted from the save.

**LE** This specifies that if the characters in the record indicated by position are less than or the same as the comparison characters, the record should be included in or omitted from the save.

## SAVE

- GT** This specifies that if the characters in the record indicated by position are greater than the comparison characters, the record should be included in or omitted from the save.
- GE** This specifies that if the characters in the record indicated by position are greater than or the same as the comparison characters, the record should be included in or omitted from the save.

### 'characters'

This specifies the comparison characters. You can specify up to 30 characters, which should be enclosed by apostrophes ('). Use any characters except apostrophes in the character string.

- REWIND** This specifies that a reel-to-reel tape is to be rewound to the load point after processing is completed. It also specifies that a tape cartridge is to be positioned to the beginning of the cartridge after processing is completed.

- LEAVE** This specifies that a reel-to-reel or cartridge tape is to be left where it was last processed after the SAVE procedure has completed. The next operation using the tape begins at that point.

- UNLOAD** This specifies that a reel-to-reel tape drive is to be rewound and unloaded after processing is completed. It also specifies that a cartridge tape is to be positioned to the end of the cartridge after processing is completed.

REWIND, LEAVE, or UNLOAD are valid only if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge is processed, or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. To do this, the tape is wound to the end, and then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, this can be a lengthy process. The preparation time can be shortened if UNLOAD is specified, because the tape is left at the end after the last operation is completed.

### NOCOMPRESS

This specifies that data is not to be compressed when a disk file is saved on diskette. The diskette file appears in the same format as the disk file from which it was copied.

### COMPRESS

This specifies that data is to be compressed when a disk file is saved on diskette. When the file is restored, the data is put back in its original format. Compressing a file usually saves space on diskette, unless the file has few or no repetitive characters.

If you specify COMPRESS, the following restrictions apply:

- You cannot use COMPRESS if you specify T1, T2, or TC.
- You cannot select or reorganize records.
- You cannot compress a disk file being added to a diskette file.

**Example 1**

The following example saves all disk files on diskette for seven days. The diskettes have volume IDs of VOL001.

```
SAVE ALL,7,,VOL001,ALL,S1,AUTO
```

**Example 2**

The following example saves a file named FILE1 and adds this file to an existing diskette file named FILE1. The volume ID of the diskette is VOL001.

```
SAVE FILE1,ADD,,VOL001
```

**Example 3**

The following example saves all files belonging to file group PAYROL. The name associated with the set of saved files is PAYROL. The volume ID of the diskette is VOL002, and the files are saved for at least 1 month (33 days). The files are compressed.

```
SAVE ALL,33,PAYROL,VOL002,PAYROL,,,,COMPRESS
```

**Example 4**

The following example saves specific records from a single file named FILE3 on a diskette. The only records saved contain the word SAVE in positions 10 through 13 of the record.

```
SAVE FILE3,,,VOL001,,,,INCLUDE,10,EQ,'SAVE'
```

**Example 5**

The following example saves all files that are not group files on tape reel VOL001, which is mounted on tape drive T1. When the reel on T1 fills, the reel on T2 is used (if the tape reel name is VOL001 and no files exist on it).

```
SAVE ALL,,,VOL001,,T1,AUTO,REWIND
```

**Example 6**

The following example saves a file named FILE1 on tape reel VOL001 mounted on tape drive T2. If additional tape reels are required to hold the file, the system only uses tape drive 2. Also, the tape does not rewind after saving the file.

```
SAVE FILE1,,,VOL001,T2,NOAUTO,,,,,LEAVE
```

**Example 7**

The following example saves a disk file named FILE1 to tape cartridge volume TEST1. The tape file is a permanent file.

```
SAVE FILE1,999,,TEST1,TC
```

---

## SAVEEXTN Procedure

Not supported. On System/36, the SAVEEXTN procedure saved to diskette all or part of the extended character file from disk. The SAVEEXTN procedure was only supported for the double-byte character set (DBCS) version of the SSP.

To do a similar function, use the Copy DBCS Font Table (CPYIGCTBL) command. The CPYIGCTBL command is only supported for the DBCS version of the OS/400 operating system.

See the *CL Reference* book for more information on the CPYIGCTBL command.  
 See the *ADTS/400: Character Generator Utility* book for more information on DBCS font tables.

## SAVEFLDR Procedure

Not supported. On System/36, the SAVEFLDR procedure saved the entire contents of a folder or all folders on the system on diskette, tape, or tape cartridge. The SAVEFLDR procedure could also save the entire contents of a single folder to a disk file.

To do a similar function, use the Save Document Library Object (SAVDLO) command with the DLO(\*ALL) parameter and the FLR(name) parameter specified.

See the *CL Reference* book for more information on the SAVDLO command. See the *System/36 Environment Programming* book for more information on using folders in the OS/400 System/36 environment. See the *Planning for and Setting Up OfficeVision/400* book for more information on folder management on the AS/400 system.

## SAVELIBR Procedure

The SAVELIBR procedure saves the entire contents of a library on diskette, tape, or tape cartridge. The procedure saves all the members of the library, together with the size of the library and the size of the library's directory. If the library contains IBM-supplied members, those members are also saved.

To restore a library copied onto diskette, tape, or tape cartridge by the SAVELIBR procedure, see the "RESTLIBR Procedure" on page 4-203.

You can copy one or more library members to diskette, tape, or tape cartridge by using the FROMLIBR procedure, but only the members are copied. The library and directory sizes are not saved. See the "FROMLIBR Procedure" on page 4-103 for more information about copying only library members.

Libraries saved using the SAVELIBR procedure cannot be restored on System/32, System/34, or System/36. Use the Save System/36 Library Members (SAVS36LIBM) command to save source and procedure members you want to restore to System/32, System/34, or System/36.

The SAVELIBR procedure runs the \$MAINT utility program.

```

SAVELIBR [ library name
           current library ], [ retention days ], volume id, [ S1 ], [ AUTO
                                                                NOAUTO ], [ I1
                                                                T1
                                                                T2
                                                                TC ],
[ REWIND
  LEAVE
  UNLOAD ]
    
```

59020640-0

**library name**

This specifies the library to save on diskette, tape, or tape cartridge. The diskette, tape, or tape cartridge file containing the library is given the same name as the library. If you do not specify the library name, the current library is assumed.

**retention days**

This specifies the number of days the diskette, tape, or tape cartridge file containing the library is retained. Specify any number from 0 to 999. If you do not specify a retention period, 999 days are assumed. If you use a retention period of 999 days, the diskette, tape, or tape cartridge file is a permanent file. For more information on diskette, tape, or tape cartridge file retention, see "FILE OCL Statement (for Diskette Files)" on page 5-33 and the "FILE OCL Statement (for Tape Files)" on page 5-37.

**volume id**

This specifies the volume ID of the diskette, tape reel, or tape cartridge. Specify from 1 to 6 alphanumeric characters. If you do not specify a volume ID, you receive a prompt to enter a volume ID.

**S1, S2, S3, M1.nn and M2.nn**

This is supported for System/36 compatibility only. The value is not used.

**AUTO** This specifies that when the tape reel (unit T1 or T2) on the original tape drive finishes, the system switches to the other tape drive to continue processing. If the tape reel on the other tape drive is not the same density as on the original tape reel, an error message is sent. If the other tape drive is not available, the system uses the original tape drive.

**NOAUTO** This specifies that when the original tape drive finishes (unit T1, T2, or TC), the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is tape, and neither AUTO nor NOAUTO are specified, AUTO is assumed.

If TC is specified for the unit, the AUTO/NOAUTO parameter is ignored.

If the unit is diskette (I1), the AUTO/NOAUTO parameter is supported for compatibility only. If specified, the value is syntax-checked, but is ignored when the diskette is processed.

**I1** This specifies that the library is saved to diskette.

**T1, T2 and TC**

These specify that the library is saved to tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

**REWIND** This specifies that a reel-to-reel tape be rewound to the load point after processing is completed. It also specifies that a tape cartridge be positioned to the beginning of the cartridge after processing is complete.

**LEAVE** This specifies that the tape or tape cartridge is not rewound after the SAVE procedure runs. The next operation using the tape begins at that point.

**UNLOAD** This specifies that a reel-to-reel tape be rewound and unloaded after processing is completed. It also specifies that a tape cartridge be positioned to the end of the cartridge after processing is complete.

REWIND, LEAVE, or UNLOAD are valid only if the unit is tape (T1, T2, or TC).

Each time a new or different tape cartridge is processed, or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. To do this, the tape is wound to the end, and then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, this can be a lengthy process. The preparation time can be shortened if UNLOAD is specified, because the tape is left at the end after the last operation is completed.

### Example 1

The following example saves a library named MYLIB permanently on a diskette with a volume ID of VOL001:

```
SAVELIBR MYLIB,999,VOL001
```

### Example 2

The following example saves the library named PAYLIB on tape with a volume ID of PAYROL. The tapes to contain the library are located on tape drives 1 and 2, starting with tape drive 2. After the library is saved, the tape is unloaded.

```
SAVELIBR PAYLIB,,PAYROL,,AUTO,T2,UNLOAD
```

---

## SAVENRD Procedure

Not supported. On the System/36, the SAVENRD procedure saved the network resource directory (NRD) on diskette, tape, or tape cartridge.

On the AS/400 system, the information that was stored in the NRD is kept in OS/400 distributed data management (DDM) files. Each DDM file is the equivalent of one NRD entry.

To do a similar function, use the Save Object (SAVOBJ) command with the OBJTYPE(\*FILE) parameter specified to save your DDM files.

See the *CL Reference* book for more information on the SAVOBJ command. See the *Distributed Data Management* book for more information on using DDM files.

---

## SDA Procedure

The SDA procedure starts the screen design aid (SDA) program. You can use SDA to:

- Create or change menus or display formats
- Help create RPG programs
- Edit library members using source entry utility (SEU)
- View display formats
- Print display formats
- Generate display files using the \$\$FGR utility program

SDA is part of the Application Development Tools licensed program.

For more information on how to use SDA, and about display formats and menus, see the *ADTS/400: Screen Design Aid for the System/36 Environment* book.

SDA	[ format member name menu name ]	[ input library name current library ]	[ N Y PARTIAL ]
	[ output library name input library ]	[ display format load member library name input library ]	

59020229-0

The parameters for the SDA procedure are optional and are used to establish the default values that appear on subsequent displays shown by SDA.

#### **format member name or menu name**

This specifies the display format source member or menu you want to create to change. Specify up to 8 characters for the format-member name and up to 6 characters for the menu name.

#### **input library**

This specifies the name of the library that contains or will contain the display format or menu source members. This parameter also specifies the library to search for display formats contained in a load member. If you do not specify this parameter, the current library is used.

Position 3 is for compatibility with the IBM System/34. This position is ignored.

- N** This specifies that only error messages, and the lines containing the errors, list when display formats generate. This is the default.
- Y** This specifies that the entire display format source member, together with any errors, is listed when display formats are generated.
- PARTIAL** This specifies that a partial listing of the display format source member lists when display formats generate.

#### **output library**

This specifies the name of the library that contains the created or changed menu, or the source for the created or changed display format. If you do not specify a library name, the input library is assumed.

#### **display file**

This specifies the name of the library that contains the created or changed display file. If you do not specify a library name, the input library is assumed.

#### **Example**

The following example shows how to start SDA:

```
SDA
```

---

## **SDALOAD Procedure**

Not supported. On System/36, the SDALOAD procedure installed the screen design aid (SDA) support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## SDASAVE Procedure

Not supported. On System/36, the SDASAVE procedure saved the screen design aid (SDA) support to diskette. See “Managing Licensed Programs” on page 4-2 for information on saving licensed programs.

---

## SECDEF Procedure

Not supported. On System/36, the SECDEF procedure allowed you to do the following:

- Create or remove the user identification file.
- Create or remove the resource security file.
- Activate or deactivate password security.
- Activate or deactivate badge security.
- Activate or deactivate resource security.
- Start or stop password date checking.

On the AS/400 system, security information is kept as part of each object, in user profiles, and authorization lists.

To do a similar function, use the Change System Value (CHGSYSVAL) command specifying SYSVAL(QSECURITY) and the level of security you want your system to have (VALUE parameter).

See the *CL Reference* book for more information on CHGSYSVAL command. See the *System/36 Environment Programming* book and the *Security – Reference* book for more information on AS/400 security.

---

## SECEDIT Procedure

Not supported. On System/36, the SECEDIT procedure allowed you to add, remove, or update entries in the user identification file or in the resource security file.

On the AS/400 system, security information is kept as part of each object, in user profiles, and authorization lists.

A similar function is available from the Security (SECURITY) menu.

To do a similar function to the System/36 for location profiles, the following commands are available on the AS/400 system:

- For APPN, use the Create Configuration List (CRTCFGL) command or the Change Configuration List (CHGCFGL) command, specifying \*APPNRMT for the configuration list type (TYPE) parameter.
  - If running the command interactively, use the default of \*PROMPT for the APPN remote location entry (APPNRMTE) parameter. A prompt screen will appear with columns to enter location password and secure location values.
  - If running the command in batch, you can specify the location password and secure location values on the APPN remote location entry APPNRMTE parameter.

- For APPC, use the Create Device Description (APPC) (CRTDEVAPPC) command on the Change Device Description (APPC) (CHGDEVAPPC) command, specifying the location password (LOCPWD) and secure location (SECURELOC) parameters.

See the *Security – Reference* book and the *System/36 Environment Programming* book for more information on AS/400 security.

See the *CL Reference* book for more information on the CRTCFGL, CHGCFGL, CRTDEVAPPC, and CHGDEVAPPC commands. See the *System/36 Environment Programming* book and the *APPC Programming* book for more information on the AS/400 communications configuration.

---

## SECLIST Procedure

Not supported. On System/36, the SECLIST procedure provided a listing of the user identification file and a listing of the resource security file.

On the AS/400 system, security information is kept as part of each object, in user profiles, and authorization lists.

To do a similar function, use the Display Object Authority (DSPOBJAUT) command, the Display User Profile (DSPUSRPRF) command, and the Display Authorization List (DSPAUTL) command. These commands are available from the Security (SECURITY) menu.

You can use the Work with Objects (WRKOBJ) command, the Work with User Profiles (WRKUSRPRF) command, and Work with Authorization Lists (WRKAUTL) command to display lists of objects, user profiles, or authorization lists. From the list display, you can select options to display the same security information provided by the DSPOBJAUT, DSPUSRPRF, and DSPAUTL commands.

To do a similar function to the System/36 for location profiles, the following commands are available on the AS/400 system:

- For APPN, use the Create Configuration List (CRTCFGL) command or the Change Configuration List (CHGCFGL) command, specifying \*APPNRMT for the configuration list type (TYPE) parameter.
  - If running the command interactively, use the default of \*PROMPT for the APPN remote location entry (APPNRMTE) parameter. A prompt screen will appear with columns to enter location password and secure location values.
  - If running the command in batch, you can specify the location password and secure location values on the APPN remote location entry APPNRMTE parameter.
- For APPC, use the Create Device Description (APPC) (CRTDEVAPPC) command on the Change Device Description (APPC) (CHGDEVAPPC) command, specifying the location password (LOCPWD) and secure location (SECURELOC) parameters.

See the *CL Reference* book for more information on the DSPOBJAUT, DSPUSRPRF, DSPAUTL, WRKOBJ, WRKUSRPRF, and WRKAUTL commands. See the *Security – Reference* book and the *System/36 Environment Programming* book for more information on AS/400 security.

## SERVICE

See the *CL Reference* book for more information on the CRTCFGL, CHGCFGL, CRTDEVAPPC, and CHGDEVAPPC commands. See the *System/36 Environment Programming* book and the *APPC Programming* book for more information on AS/400 communications configuration.

---

### SECRET Procedure

Not supported. On System/36, the SECRET procedure restored the user identification file or the resource security file with the copy saved by the SECSAVE procedure.

On the AS/400 system, security information is kept as part of each object, in user profiles, and authorization lists.

To do a similar function, use the Restore User Profile (RSTUSRPRF), Restore Object (RSTOBJ), and Restore Authority (RSTAUT) commands.

See the *CL Reference* book and the *Backup and Recovery – Advanced* book for more information on the RSTUSRPRF, RSTOBJ, and RSTAUT commands. See the *Security – Reference* book and the *System/36 Environment Programming* book for more information on AS/400 security.

---

### SECSAVE Procedure

Not supported. On System/36, the SECSAVE procedure saved the user identification file or the resource security file on diskette, disk, tape, or tape cartridge.

On the AS/400 system, security information is kept as part of each object, in user profiles, and authorization lists.

To do a similar function, use the Save System (SAVSYS) command.

See the *CL Reference* book and the *Backup and Recovery – Advanced* book for more information on the SAVSYS command. See the *Security – Reference* book and the *System/36 Environment Programming* book for more information on AS/400 security.

---

### SERVICE Procedure

Not supported. On System/36, the SERVICE procedure displayed a menu that allowed you to perform various problem determination tasks.

To do a similar function, use the Work with Problems (WRKPRB) command. This command can be selected from the Problem Handling (PROBLEM) menu.

See the *CL Reference* book for more information on the WRKPRB command. See the *System Operation* book for more information on working with system problems.



**YMD** This specifies that the session date format and the AS/400 job date format are year-month-day.

The System/36 date format is also affected by the CHGJOB CL command. For more information on handling the date format in a System/36 environment job, see the job date and date format section in the *System/36 Environment Programming* book.

**mmddy, ddmmy, or yymmdd**

This specifies the session date and the AS/400 job date. The date must be specified in the appropriate session date format.

The System/36 session date is also affected by the CHGJOB CL command. For more information on handling the session date in a System/36 environment job, see the job date and date format section in the *System/36 Environment Programming* book.

**region size**

This is supported for System/36 compatibility only. The value is not used.

**library name**

This specifies the sign-on library assigned to the display station. The system library (#LIBRARY) can be assigned. If 0 is specified, no library is active.

The specified library does not become active until the *next* time an operator signs on at the display station. For more information, see the *System/36 Environment Programming* book.

**printer id or SYSTEM**

This specifies the default sign-on printer used for system list output created from the display station.

Use the PRINT or SYSLIST procedure to immediately change the printer used for system list output. See the "PRINT Procedure" on page 4-178 or the "SYSLIST Procedure" on page 4-253 for information about running these procedures.

**printer id** specifies the 2-character ID of the printer to use as the sign-on printer for the display station.

**SYSTEM or SYS** specifies that the system printer is used as the sign-on printer for the display station.

**forms number**

This specifies the forms number to be used for display station output. You can also change the forms number by:

- A FORMS statement
- A PRINTER statement (for that print step only)
- The PRINT procedure

**print key printer id**

This parameter is not valid. If a value is specified, an error message is issued.

**BORDER or NOBORDER**

This parameter is not valid. If a value is specified, an error message is issued.

**HEADER or NOHEADER**

This parameter is not valid. If a value is specified, an error message is issued.

---

**SETALERT Procedure**

Not supported. On System/36, the SETALERT procedure changed the alert indicators for messages in a message load member. Both IBM-supplied messages and user-defined messages were assigned these values. The setting of the alert indicator determined whether an alert would be sent when the message was issued.

On the AS/400 system, all messages contain an alert option.

To do a similar function, use the Change Message Description (CHGMSGD) command, specifying a value for the alert options (ALROPT) parameter for each message that you want to generate an alert message.

See the *CL Reference* book for more information on the CHGMSGD command. See the *Alerts Support* book for more information on specifying and generating alerts. See the *System/36 Environment Programming* book for more information on messages in the System/36 environment.

---

**SETCOMM Procedure**

Not supported. On System/36, the SETCOMM procedure allowed you to set certain communications items in the communications configuration record.

To do a similar function, use the Work with Line Descriptions (WRKLIND), Work with Device Descriptions (WRKDEVD), or Work with Controller Descriptions (WRKCTLD) commands. From these *Work with* displays, you can change or create the associated descriptions. You can use the Work with Configuration Status (WRKCFGSTS) command to determine the status of the line or device you want to change.

On the AS/400 system, not all line characteristics can be changed. If an application requires a line with multiple variations of these characteristics, create multiple line descriptions and vary on the line description needed.

The following is a mapping of the System/36 SETCOMM procedure parameters to the equivalent OS/400 command parameters:

**line number**

On System/36, this specified the number of the communications line for which the settings were to be changed.

On the AS/400 system, specify the line description (LIND) parameter on the appropriate command for changing the line description.

**NONSWTCH**

On System/36, this specified that the line was a point-to-point non-switched line.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

**SWITCHED**

On System/36, this specified that the line is a point-to-point switched line.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

**MULTTRIB**

On System/36, this specified that the System/36 was a multipoint tributary station.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

**MULTCONT**

On System/36, this specified that the System/36 is a multipoint control station. This parameter cannot be used with BSC.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need. This parameter is not supported for BSC.

**SHM**

On System/36, this specified that the line used X.21 short hold mode, which helped reduce line usage by disconnecting the circuit-switched line when there was no line activity.

On the AS/400 system, this is a parameter that cannot be changed in the line description. Create a separate line description for each line type that you need.

**CLOCK**

On System/36, this specified that the system must provide business machine clocking for data communications.

On the AS/400 system, this function is not supported.

**NOCLOCK**

On System/36, this specified that the modem or another external source had the clocking facility.

On the AS/400 system, this function is not supported.

**NRZI**

On System/36, this specified that NRZI data encoding was performed whenever the system was using SDLC protocol for modems that were sensitive to certain bit patterns in the data stream. NRZI could only be used if the System/36 modem, the remote system modem, and the remote system or device were also using NRZI.

On the AS/400 system, specify \*YES on the encoding parameter (NRZI) of the Change Line Description (SDLC) (CHGLINSDLC) command.

**NONRZI**

On System/36, this specifies that the line was non-NRZI.

On the AS/400 system, specify \*NO on the encoding (NRZI) parameter of the Change Line Description (SDLC) (CHGLINSDLC) command.

**CONCAR**

On System/36, specified continuous carrier. This was a feature that held the 'request-to-send' signal active. CONCAR was specified when modems or modem eliminators were used and the multipoint control station was on a four-wire nonswitched communications facility, or when

the interface was a four-wire nonswitched point-to-point communications facility and the modem did not support switched network backup.

On the AS/400 system, specify the duplex (DUPLEX) parameter of the Change Line Description (SDLC) (CHGLNSDLC) command. CONCAR was also specified if an X.25 network was used.

#### **NOCONCAR**

On System/36, this specified that the continuous carrier feature would not be used. NOCONCAR was specified for a multipoint tributary station, for 2-wire point-to-point networks, for a switched public telephone line, or for an interface that used a Digital data service adapter (DDSA) or used an X.21 interface adapter on a switched network.

On the AS/400 system, specify the frame retry count (FRAMERTY) parameter of the Change Line Description SDLC (CHGLNSDLC) command. Other retry values can also be specified on the CHGLNSDLC command.

**tone** On System/36, this specified that a non-United States answer tone was required for manual answer and autoanswer communications. This setting was modem dependent.

On the AS/400 system, this function is not supported.

**NOTONE** On System/36, this specified that a non-United States answer tone was not required.

On the AS/400 system, this function is not supported.

**SEP** On System/36, specified that your autocal unit supports separator characters. When the system attempted to dial a phone number containing a separator character, the character was sent to the autocal unit to do the separation delay.

On the AS/400 system, this function is not supported.

**NOSEP** On System/36, this specified that your autocal unit did not support separator characters. When the system attempted to dial a phone number containing a separator character, the character was not sent to the autocal unit. Instead, the system did a 3 second separation delay.

On the AS/400 system, this function is not supported.

**EON** On System/36, this specified that your autocal unit did not have answer tone detection capability. If your autocal unit did not have answer tone detection capability, you should have used an end-of-number character in the phone numbers you specified with the DEFINEPN procedure. The end-of-number character was sent from the system to the autocal unit.

On the AS/400 system, this function is not supported.

**NOEON** On System/36, this specified that your autocal unit had answer tone detection capability. If you used an end-of-number character in the phone numbers you specified with the DEFINEPN procedure, the character would not be sent from the system to the autocal unit.

On the AS/400 system, this function is not supported.

**primary sdlc time-out value**

On System/36, this specified the primary SDLC time-out value in half-second increments.

On the AS/400 system, specify the idle timer (IDLTMR) parameter of the Change Line Description (SDLC) (CHGLNSDLC) command. Other timer values can also be specified on the CHGLNSDLC command.

**sdlc retry count**

On System/36, this specified the number of primary SDLC error retries to be attempted.

On the AS/400 system, specify the frame retry count (FRAMERTY) parameter of the Change Line Description (SDLC) (CHGLNSDLC) command. Other retry values can also be specified on the CHGLNSDLC command.

**IBMLPDA**

On System/36, this specified an IBM modem with link problem determination aid (LPDA) functions. This included IBM external modems such as the 3833, 3834, 3863, 3864, 3865, 3868, 5812, 5865, 5866, and 5868.

On the AS/400 system, specify \*IBMLPDA1 or \*NORMAL for the modem type supported (MODEM) parameter on the appropriate change line description command.

**IBMWRAP**

On System/36, specified an IBM modem with wrap test capabilities. This included IBM external modems such as the 3872, 3874, 3875, 5811, 5841, and 5842.

On the AS/400 system, specify \*IBMWRAP or \*NORMAL for the modem type supported (MODEM) parameter on the appropriate change line description command.

**NONIBM** On System/36, this specified non-IBM modems being used with EIA/CCITT interface adapters.

**Note:** Most IBM modems ran either the IBM LPDA tests or the IBM wrap tests. All other modems were configured to the System/36 as non-IBM modems. Refer to the modem books for information on the tests that are supported.

On the AS/400 system, specify \*NORMAL for the modem type supported (MODEM) parameter on the appropriate change line description command.

**X25 or NOX25**

On System/36, this specified whether the line used X.25 support.

On the AS/400 system, specify the link type (LINKTYPE) parameter on the create controller description commands that support X.25 (CRTCTLAPPC, CRTCTLASC, CRTCTLFNC, CRTCTLRWS, and CRTCTLHOST).

**2400BPS** On System/36, this specified a DDSA (Digital Data Service Adapter/Attachment) line with a line speed of 2400 bps.

On the AS/400 system, this function is not supported.

**4800BPS** On System/36, this specified a DDSA line with a line speed of 4800 bps.  
On the AS/400 system, this function is not supported.

**9600BPS** On System/36, this specified a DDSA line with a line speed of 9600 bps.  
On the AS/400 system, this function is not supported.

**56KBPS** On System/36, this specified a DDSA line with a line speed of 56 000 bps.  
On the AS/400 system, this function is not supported.

#### **secondary SDLC inactivity time-out value**

On System/36, this specified the time period that determined when the primary system was to be considered no longer active for a secondary SDLC nonswitched line. You could specify a number from 1 through 20 to indicate the number of 32-second multiples for the timer, or you could specify 0 to indicate that no secondary inactivity timer should be used.

On the AS/400 system, specify the inactivity timer (INACTTMR) parameter of the Change Line Description SDLC (CHGLNSDLC) command. Other timer values can also be specified on the CHGLNSDLC command.

#### **token-ring network adapter address override**

On System/36, this specified that you wanted to replace the universally-administered adapter address (preset address) with a locally-administered address (address controlled by you), or that you wanted to reset the locally-administered address to the universally-administered address.

On the AS/400 system, specify the local adapter address (ADPTADR) parameter of the Change Line Description (Token-Ring Network) (CHGLINTRN) command.

#### **XXXXXXXXXXXX**

The locally-administered adapter address replacing the universally-administered address. The specified hexadecimal address must be within the 400000000000 - 7FFFFFFFFFFF range. On the AS/400 system, this range of values is the same.

**R** Reset the locally-administered address to the universally-administered address. On the AS/400 system, specify \*ADPT for the ADPTADR parameter.

See the *CL Reference* book for more information on the CHGLINTRN, CRTCTLHOST, CRTCTLRWS, CRTCTLFNC, CRTCTLAPPC, CRTCTLASC, CHGLNSDLC, WRKLIND, WRKDEVD, WRKCTLD, and WRKCFGSTS commands. See the *System/36 Environment Programming* book for more information on AS/400 communications configuration.

---

## SETDUMP Procedure

Not supported. On System/36, the SETDUMP procedure allowed the debugging of a program running in main storage at predetermined break points or addresses, without having to stop the main storage processor. The SETDUMP procedure also allowed a task dump to be taken when an address in the program was referred to.

A similar function is available from the Debug Commands (CMDDBG) and Break-point Commands (CMDDBKP) menus.

See the *CL Programming* book for more information on the testing and debugging programs. See the *System Operation* book for more information on working with system problems.

---

## SEU Procedure

The SEU procedure starts the source entry utility (SEU) program. Use SEU to create or change programs, procedures, message members, menus, or display formats.

SEU is part of the Application Development Tools licensed program. For more information on how to use SEU, see the *ADTS/400: Source Entry Utility* book. For information about display formats and menus, see the *System Operation for New Users* book. For more information about message members, see "CREATE Procedure" on page 4-66.

```

SEU  member name, [SOURCE], [seu format member], [statement length],
                (S)
                PROC
                (P)
                A
                F
                R
                T
                W

                [library name], [diagnosed source file]
                current library

```

S9020243-0

If you did not specify any parameters, a display appears on which you can enter the parameters.

### member name

This specifies the source member or procedure that you want to create or change. Specify up to 8 characters and begin the member name with an alphabetic character (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special), except blanks, commas, and apostrophes.

Do not use commas (,), hyphens (-), apostrophes ('), question marks (?), slashes (/), greater than signs (>), plus signs (+), or equal signs (=) because these have special meanings in procedures. Do not use DIR, LIBRARY, or ALL as a member name.

### SOURCE or S

This specifies a library source member. If you do not specify a parameter, SOURCE (or S) is assumed.

**PROC or P**

This specifies a library procedure member.

- A** This specifies an RPG II program that contains auto report specifications. SEU checks the syntax of the RPG II program and displays the RPG specification for each statement in the source member.
- F** This specifies a library source member that contains display format S- and D-specifications and displays the format specification for each statement in the source member.
- R** This specifies a library source member that contains an RPG II program. SEU checks the syntax of the RPG II program and displays the RPG specification for each statement in the source member.
- T** This is supported for System/36 compatibility only. The value is not used.
- W** This specifies a source member containing work station utility (WSU) statements. SEU displays the WSU specification for each statement in the source member.

**SEU format member**

This is supported for System/36 compatibility only. The value is not used.

**statement length**

This is supported for System/36 compatibility only. The value is not used.

**library name**

This specifies the library that contains or will contain the library member being changed or created. If you do not specify a library name, the current library is assumed.

**diagnosed source file**

This is supported for System/36 compatibility only. The value is not used.

**Example**

The following example starts SEU to create or change a procedure named PAYROLL, contained in the current library:

```
SEU PAYROLL,P
```

---

**SEULOAD Procedure**

Not supported. On System/36, the SEULOAD procedure installed the source entry utility (SEU) support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

**SEUSAVE Procedure**

Not supported. On System/36, the SEUSAVE procedure saved the source entry utility (SEU) support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## SHRFLOAD Procedure

Not supported. On System/36, the SHRFLOAD procedure installed the shared folders facility support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## SHRFSAVE Procedure

Not supported. On System/36, the SHRFSAVE procedure saved the shared folders facility support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

## SLIB Procedure

The SLIB procedure specifies the name of the library for the display station session. The session library and the current library are the same when you are entering procedures or control commands from the keyboard. Change the current library for a procedure by using the LIBRARY OCL statement. See "LIBRARY OCL Statement" on page 5-54 for more information.

The current library is the library that is searched first for procedures, programs, menus, display formats, and message members. If the member is not found in the current library, the system library and the library list are searched.

A library remains the session library until you sign off the system or do one of the following:

- Enter another SLIB procedure.
- Process or enter the MENU OCL statement or command with a library parameter.
- Enter a LIBRARY OCL statement.
- Process a LIBRARY OCL statement with SESSION-YES in a procedure.

Once you specify a session library, that library remains allocated to the display station until you specify a different library or the system library.

A program within a procedure can be interrupted by a system request. The sign-on that follows is to a new job. Neither the current nor the session library of the interrupted job is used in the new job.

If you enter the SLIB procedure from a MENU display, one or more of the items on the displayed menu might not be correct because the procedures corresponding to those items do not exist in the new session library. Therefore, you should not change libraries without changing the menu.

The SLIB procedure processes a LIBRARY OCL statement with SESSION-YES specified. See "LIBRARY OCL Statement" on page 5-54 for more information.

```
SLIB    library name
```

59020246-0

**library name**

This specifies the library that will be the session library. You must specify a library name.

**Example**

The following example specifies MYLIB as the session library:

```
SLIB MYLIB
```

---

## SMF Procedure

Not supported. On System/36, the SMF procedure allowed you to record and print information about how your system was being used. The SMF procedure displayed a menu that allowed you to select the SMF task you wanted to do.

To do a similar function, use the Start Performance Tools (STRPFRT) command. The STRPFRT command is part of the Performance Tools/400 licensed program.

See the *CL Reference* book for more information on the STRPFRT command. See the *Performance Tools/400* book for more information on the Performance Tools/400 licensed program.

---

## SMFDATA Procedure

Not supported. On System/36, the SMFDATA procedure was part of the system measurement facility (SMF). The SMFDATA procedure wrote output for an SMF report to a disk file that could be used as input for analysis by application programs.

On the AS/400 system, a similar function is available from the OS/400 Performance Monitor. The files created by the Performance Monitor may be accessed by a user program or through the Performance Tools/400 licensed program.

See the *Work Management* book for more information on collecting performance data and the layout of the files created by the Performance Monitor. See the *Performance Tools/400* book for more information on the Performance Tools/400 licensed program.

---

## SMFPRINT Procedure

Not supported. On System/36, the SMFPRINT procedure listed a formatted report of the data collected by the system measurement facility (SMF) data collection program.

To do a similar function, use the Start Performance Tools (STRPFRT) command, and select option 3 (Select performance reports). Reports are provided that range from a high-level view of the system down to a detailed analysis of individual jobs, programs, or transactions. The STRPFRT command is part of the Performance Tools/400 licensed program.

See the *CL Reference* book for more information on the STRPFRT command. See the *Work Management* book for more information on collecting performance data.

See the *Performance Tools/400* book for more information on the Performance Tools/400 licensed program.

---

### SMFSTART Procedure

Not supported. On System/36, the SMFSTART procedure started the system measurement facility (SMF) data collection program.

To do a similar function, use the Start Performance Monitor (STRPFRMON) command. The files created by the Performance Monitor may be accessed by a user program or through the Performance Tools/400 licensed program.

See the *CL Reference* book for more information on the STRPFRMON command. See the *Work Management* book for more information on collecting performance data and the layout of the files created by the Performance Monitor. See the *Performance Tools/400* book for more information on the Performance Tools/400 licensed program.

---

### SMFSTOP Procedure

Not supported. On System/36, the SMFSTOP procedure stopped the system measurement facility (SMF) data collection program.

To do a similar function, use the End Performance Monitor (ENDPFRMON) command. The files created by the Performance Monitor may be accessed by a user program or through the Performance Tools/400 licensed program.

See the *CL Reference* book for more information on the ENDPFRMON command. See the *Work Management* book for more information on collecting performance data and the layout of the files created by the Performance Monitor. See the *Performance Tools/400* book for more information on the Performance Tools/400 licensed program.

---

### SOFTWARE Procedure

Not supported. On System/36, the SOFTWARE procedure allowed you to build a tape or diskette of software for redistribution to remote System/36 systems licensed for that software.

A similar function is available from the Manage Licensed Programs (LICPGM) menu for managing IBM licensed programs. On the AS/400 system, you can only use tape for saving and restoring IBM licensed programs. No similar function is available on the AS/400 system for managing non-IBM software.

See "Managing Licensed Programs" on page 4-2 for information on managing licensed programs. See the *Software Installation* book for more information on managing licensed programs.

---

## SORT Procedure

The SORT procedure allows you to sort data in disk files. The actual sorting is done by the Reformat Data (FMTDTA) command.

```

SORT   input file name,source member name,output file name,
       number of records, [ source member library ] , [ N ]
                          [ current library       ] , [ Y ]

```

S9020252-0

### input file name

This specifies the file to sort. The input file is located in the current System/36 environment files library.

### source member name

This specifies the source member that contains the specifications used to sort the file.

### output file name

This specifies the file that contains the sorted records. The output file is located in the current System/36 environment files library.

### number of records

This specifies the number of records to allocate to the output file. The number can be from 1 to 8000000.

### source member library

This specifies the name of the library that contains the source member. If you do not specify a library name, the current library is assumed.

**N** This specifies that the job is run from the display station that requested the procedure, not placed on the job queue. If you do not specify a parameter, N is assumed.

**Y** This specifies that the job is run from the job queue.

### Example

The following example sorts a file named FILE1. The sorted output file is named FILE2. The sort specifications are in a source member named SORTNAME in the library named MYLIB. FILE2 contains up to 200 records.

```
SORT FILE1,SORTNAME,FILE2,200,MYLIB
```

---

## SPECIFY Procedure

The IBM System/34 SPECIFY procedure is not supported. On System/36, the user was referred to the ALTERCOM procedure.

Similar function is available from the Configure Devices and Communications (CFGDEVCMN) menu. From this menu you can change communications configuration information.

See the *Local Device Configuration* book for more information on communications and device configuration.

## SRTX Procedure

The SRTX procedure allows you to sort the following double-byte character set (DBCS) character sets:

- Japanese
- Korean
- Traditional Chinese

The actual sorting is performed by the Reformat Data (FMTDTA) command. SRTX uses active sort tables QCGACTV, QCGACTVK, and QCGACTVC, in conjunction with the information provided in the sort specifications to determine the correct sort sequence. For more information on DBCS sort tables and sorting DBCS characters, refer to the *ADTS/400: Character Generator Utility* book, and the *Data Management* book.

```
SRTX  input file name,source member name,output file name,
      number of records, [ source member library ] , [ N ]
                        [ current library ]         [ Y ]
```

59020253-0

### input file name

This specifies the file to sort. The input file is normally located in library QS36F.

### source member name

This specifies the library source member that contains the specifications used to sort the file.

### output file name

This specifies the file that contains the sorted records. The output file is normally located in library QS36F.

### number of records

This specifies the number of records to allocate to the output file. The number can be from 1 to 8000000.

### source member library

This specifies the name of the library that contains the source member. If you do not specify a library name, the current library is assumed.

**N** This specifies that the job is run from the display station that requested the procedure, not placed on the job queue. If you do not specify a parameter, N is assumed.

**Y** This specifies that the job is to be run from the job queue.

### Example

The following example sorts a file named FILE1. The sorted output file is named FILE2. The sort specifications are in a source member named IDEOGRPH in the MYLIB library. FILE2 contains 200 records.

```
SRTX FILE1,IDEOGRPH,FILE2,200,MYLIB
```

---

## SRTXBLD Procedure

The SRTXBLD procedure updates the QCGACTV or QCGACTVC DBCS active sort table (was the #KACTIVE file on System/36) from the QCGMSTR or QCGMSTRC DBCS master sort table (was the #KAMAST file on System/36) so the DBCS active collating sequence will reflect any changes made to DBCS characters after using the character generator utility (CGU). You should also use this procedure after migrating your master file (#KAMAST) from the System/36 to the AS/400 system.

**Note:** The SRTXBLD procedure supports both the Japanese (QCGACTV) and the Traditional Chinese (QCGACTVC) collating sequences. To update either of these active sort tables you must be on a DBCS display configured for the appropriate language. For example, to update the Japanese table you must be on a DBCS display configured for Japanese. To use the SRTXBLD procedure, you must have installed the character generator utility (part of the Application Development Tools licensed program). For more information on DBCS sort tables and sorting DBCS characters, refer to the *ADTS/400: Character Generator Utility* book.



The SRTXBLD procedure has no parameters.

---

## SRTXLOAD Procedure

Not supported. On System/36, the SRTXLOAD procedure installed the DBCS sort support from diskette. See “Managing Licensed Programs” on page 4-2 for information on installing licensed programs.

---

## SRTXSAVE Procedure

Not supported. On System/36, the SRTXSAVE procedure saved the DBCS sort support to diskette. See “Managing Licensed Programs” on page 4-2 for information on saving licensed programs.

---

## STARTM Procedure

Not supported. On the System/36, the STARTM procedure was used to start automatic monitoring of a BSC multipoint line for a particular station address.

On the AS/400 system, automonitor support for a BSC multipoint tributary station and location address becomes active after the line, controller and device descriptions for that particular station and location address pair are successfully varied on.

**Note:** On System/36, only a station address was specified for the STARTM procedure. On the AS/400 system, a station address must be specified in the line description and a location address must be specified in the device description.

## STATEST

To do a similar function, use the Vary Configuration (VRYCFG) command. To start monitoring a particular station and location address pair, vary on the line, controller and device description associated with the pair. You may optionally vary on more than one device description for a particular station address. Each device description must contain a unique location address. You can use the Create Line Description BSC (CRTLINBSC), Create Controller Description BSC (CRTCTLBSC), and Create Device Description BSC (CRTDEVBSC) commands to create these descriptions.

See the *CL Reference* book for more information on the VRYCFG, CRTLINBSC, CRTCTLBSC, and CRTDEVBSC commands. See the *System/36 Environment Programming* book for more information on AS/400 communications configuration. See the *BSC Equivalence Link Programming* book for more information on AS/400 BSC support.

---

## STATEST Procedure

Not supported. On System/36, the STATEST procedure tested the communications line and the controller used for remote display stations and printers. It also tested SSP-ICF Finance controllers and SSP-ICF secondary Peer connections. (System/36 was the primary station for APPC or SSP-ICF Peer.)

A similar function is available by running the link test function provided by the Verify Communications (VFYCMN) command. Use the link test function to send an architected message to a remote controller.

The VFYCMN command prompts for the line name to be tested. The next display that appears allows you to select the communications test to be run. When you select the link test function, a series of displays appear that allow you to choose how many repetitions of the test message are to be sent and which test pattern is to be sent. The test patterns differ by protocol and can be selected from a list.

When running the link test, consider the following:

- Only one controller can be tested per run.
- Asynchronous communications are not supported.
- The wait time between messages cannot be specified.
- Choosing whether to log successful messages is not supported.
- For SDLC, the primary line must be used.
- For X.25, either the primary channel or a negotiable channel that negotiates to the primary must be used.
- For APPC devices, a test can be sent over both switched and nonswitched lines, unlike the capabilities on the System/36.
- A station must be configured to be tested.

See the *CL Reference* book and the *AS/400 Licensed Internal Code Diagnostic Aids – Volume 1* book for more information on the VFYCMN command.

---

## STOPGRP Procedure

Not supported. On System/36, the STOPGRP procedure stopped a single session group or all session groups configured for a remote location using the advanced program-to-program communications (APPC) or advanced peer-to-peer networking (APPN) subsystems.

To do a similar function, use the End Mode (ENDMOD) command. The ENDMOD command stops all modes when the current activity stops. The STOPGRP immediate function is not supported by the ENDMOD command. When you use the End Job (ENDJOB) command, specify \*IMMED for the how to end (OPTION) parameter to end any active jobs. Then use the Vary Configuration (VRYCFG) command to end the mode.

See the *CL Reference* book for more information on the ENDMOD, ENDJOB, and VRYCFG commands. See the *APPC Programming* book for more information on APPC and APPN. See the *System/36 Environment Programming* book for more information on AS/400 communications configuration.

---

## STOPM Procedure

Not supported. On System/36, the STOPM procedure stopped the automatic monitoring function of a BSC multipoint line.

On the AS/400 system, automonitor support for a BSC multipoint tributary station and location address pair stops after the device description for that particular station and location address pair is varied off.

To do a similar function, use the Vary Configuration (VRYCFG) command. To stop monitoring a particular location address, vary off the device description for a particular station and location address pair. To stop monitoring a particular BSC multipoint tributary line, vary off all device descriptions associated with the varied on line description that contains the station address for that line.

See the *CL Reference* book for more information on the VRYCFG commands. See the *System/36 Environment Programming* book for more information on the AS/400 communications configuration.

---

## STRTPGRP Procedure

Not supported. On System/36, the STRTPGRP procedure started a single session group or all session groups configured for a remote location using the advanced program-to-program communications (APPC) or advanced peer-to-peer networking (APPN) subsystems.

To do this function, use the Start Mode (STRMOD) command.

See the *CL Reference* book for more information on the STRMOD command. See the *APPC Programming* book and the *APPN Support* book for more information on APPC and APPN. See the *System/36 Environment Programming* book for more information on AS/400 communications configuration.

---

## SWDLOAD Procedure

Not supported. On System/36, the SWDLOAD procedure installed the software distribution support from diskette. See “Managing Licensed Programs” on page 4-2 for information on installing licensed programs.

---

## SWDSAVE Procedure

Not supported. On System/36, the SWDSAVE procedure saved the software distribution support to diskette. See “Managing Licensed Programs” on page 4-2 for information on saving licensed programs.

---

## SWITCH Procedure

The SWITCH procedure sets one or more of the user program status indicator (UPSI) switches for the display station to on (1) or off (0). The switch setting remains in effect until the display station session ends or one of the following occurs:

- Another SWITCH procedure processes.
- A SWITCH OCL statement processes.
- A program changes the setting of any of the indicators.

All switches are set to off when a display station session begins (when an operator signs on).

A job placed on the job queue uses a copy of the switches for the display station as the switches existed when the job was placed on the queue.

Use the STATUS SESSION command to determine the current switch settings. For information on determining the switch settings in a procedure, see “SWITCH (Switches) Condition” on page 3-39.

### Notes:

1. If an SSP procedure changes the setting of a switch, the switch is returned to its original setting when the SSP procedure ends.
2. A set of switches exists for each running multiple requester terminal (MRT) procedure.

The SWITCH procedure processes a SWITCH OCL statement.

```
SWITCH  switch settings
```

59020262-0

### Switch settings

This specifies how the switches are set. Switch settings consist of 8 characters, one for each of the eight UPSI switches. The first character from the left gives the setting of switch 1, the second character gives the setting of switch 2.

The parameter always must contain 8 characters. For each switch, you must use one of the following characters:

Character	Meaning
0 (zero)	Set the switch off.
1 (one)	Set the switch on.
X	Leave the switch as it is.

### Example

This example sets the switches to the following:

Switch	Switch Setting	Result
1	1	Set on
2	X	Unaffected
3	0	Set off
4	1	Set on
5	1	Set on
6	0	Set off
7	X	Unaffected
8	X	Unaffected

```
SWITCH 1X0110XX
```

## SYSLIST Procedure

The SYSLIST procedure causes the system list output to be handled in one of the following ways:

- Listed on the printer that was assigned to the display station during system configuration
- Listed on one of the other printers
- Displayed at the display station
- Not listed

Use the PRINT procedure to have all your printed output go to a specified printer. For more information, see “PRINT Procedure” on page 4-178.

The SYSLIST assignment remains in effect until you sign off the system or the assignment is changed by one of the following:

- Another SYSLIST procedure
- The PRINT procedure
- A SYSLIST OCL statement

**Note:** The SYSLIST procedure does not change the system list device if debugging is turned on. For more information, refer to “DEBUG OCL Statement” on page 5-22.

The SYSLIST procedure processes a SYSLIST OCL statement.

```
SYSLIST [ PRINTER ] , [ EXTN ] , [ FOLD ]
         [ CRT ]      [ NOEXTN ] [ NOFOLD ]
         [ printer id ]
         [ OFF ]
```

59020263.0

- PRINTER** This specifies that system list output lists on the printer assigned to the display station during system configuration by the SET procedure or by the \$SETCF utility program. If you do not specify a parameter, PRINTER is assumed.
- CRT** This specifies that system list output displays at the display station.  
After each system list display, the operator can request that the next set of lines is displayed or that the display ends. If the operator enters a blank, the next display shows the same number of lines shown on the previous display. If the operator enters 0, the display ends. If the operator enters characters other than a blank or 0 through 18, the next set of lines appears.
- printer id** This specifies that system list output prints on the printer with the specified printer ID. Printer IDs are assigned during System/36 environment configuration. To print output on the system printer, specify the printer ID of the system printer.
- OFF** This specifies that the System/36 environment system list output does not print or display.
- EXTN** This specifies that the extended characters print or display in the system list output.
- NOEXTN** This specifies that the extended characters in the system list output do not print or display. The system-defined default double-byte character is listed for any extended character.  
**Note:** The EXTN and NOEXTN parameters are for the double-byte character set version of the OS/400 operating system and are ignored for non-double-byte character set systems. When a session begins, extended character processing for system list output is assumed. During the session, the EXTN and NOEXTN parameters can be used to turn on and turn off extended character processing for system list output.
- FOLD** This specifies that the System/36 environment system list output appearing at the display station is not truncated if it exceeds the column width of the display station. Instead, data continues on the next line of the display.
- NOFOLD** This specifies that the System/36 environment system list output displayed at the display station is truncated if it exceeds the column width of the display station. If you do not specify a parameter, NOFOLD is assumed.

## Example

The following example prints the output on the printer assigned to the display station:

```
SYSLIST PRINTER
```

---

## SYSTYPE Procedure

The SYSTYPE procedure assigns a four digit return code that indicates which operating system is processing the procedure. The ?CD? (Return Code) Procedure Control Substitution Expression can be used to evaluate the return code value after running the procedure.

SYSTYPE

R15F002-0

The SYSTYPE procedure has no parameters.

The SYSTYPE procedure does not exist on the System/36 or on the AS/400 Advanced 36 (Model 236). You have to create the procedure in the library #LIBRARY on either of those systems in order to use it there.

- On the System/36, put the following single EVALUATE statement in the procedure:

```
// EVALUATE CD=3600
```

- On the AS/400 Advanced 36 (Model 236), put the following single EVALUATE statement in the procedure:

```
// EVALUATE CD=3602
```

### Example

In the following example, the SYSTYPE procedure is run within another procedure to determine if that procedure is running in the System/36 environment. If the return code, set by the SYSTYPE procedure, indicates that the procedure is running in the System/36 environment, the FILELIB OCL statement is used to set the current files library to a library named PAYROLL. If the return code indicates that the procedure is not running in the System/36 environment, the FILELIB OCL statement is not used. In order to run a procedure that uses the SYSTYPE procedure on a System/36 or an AS/400 Advanced 36 (Model 236), you have to create the SYSTYPE procedure on those systems, as explained above.

```
SYSTYPE
// IF ?CD?=3601 FILELIB NAME=PAYROLL,SESSION=NO
```

---

## TAPECOPY Procedure

Use the TAPECOPY procedure to do the following:

- Copy a disk file to an exchange tape file.
- Add a file from disk to an existing exchange tape file.
- Copy an exchange file from tape to a disk file.
- Add an exchange file from tape to an existing disk file.

# TAPECOPY

## Notes:

1. If nonlabeled (NL), nonstandard label (NS), or bypass label processing (BLP) is specified when copying or adding a tape file to a disk file, you are prompted for the number of tape reels to be processed. If the tape file is on a single tape reel, type 1 and press the Enter key. If the tape file occupies more than one tape reel (multivolume), type in the number of tape reels the file occupies and press the Enter key.
2. You cannot use TAPECOPY with a tape cartridge (TC) drive.

The TAPECOPY procedure runs the \$TCOPY utility program.

To copy a disk file to an exchange tape file, use the following procedure:

```
TAPECOPY label1, [mmddyy  
ddmmyy  
yyymmdd], [F1], [NOADD], [label2  
label1], [T1  
T2], [retention days],  
[NOAUTO  
AUTO], [STDLABEL  
(SL)  
NONLABEL  
(NL)], volume id, [FIXED  
(F)  
FIXEDBLK  
(FB)], [record length],  
[block length], [REWIND  
LEAVE  
UNLOAD], [sequence number]
```

59020264-0

To add a file from disk to an existing exchange tape file:

```
TAPECOPY label1, [mmddyy  
ddmmyy  
yyymmdd], [F1], {ADD  
ADDNOCHK}, [label2  
label1], [T1  
T2], [mmddyy  
ddmmyy  
yyymmdd],  
[NOAUTO  
AUTO], [STDLABEL  
(SL)  
NONLABEL  
(NL)], volume id, [FIXED  
(F)  
FIXEDBLK  
(FB)], [record length],  
[block length], [REWIND  
LEAVE  
UNLOAD], [sequence number]
```

59020265-0

To copy an exchange file from tape to a disk file:

```

TAPECOPY label1, [mmddyy, [T1], [NOADD], [label2], [F1], ,
                  [ddmmyy, [T2], [label1], [F1], ,
                  [yymmdd]

[NOAUTO], [STDLABEL], [volume id], [FIXED], [record length],
[AUTO]    (SL)          (F)
          (NL)          (FB)
          (NS)          (V)
          (BYPASS)
          (BLP)

[block length], [REWIND], [sequence number], (RECORDS, records),
               (LEAVE)
               (UNLOAD)
               (BLOCKS, blocks)

[key length], [key location], [NODUPKEY]
               (DUPKEY)
    
```

59020266-1

To add an exchange file from tape to an existing disk file:

```

TAPECOPY label1, [mmddyy, (T1), ADD, [label2], [F1], [mmddyy],
                  [ddmmyy, (T2), [label1], [F1], [ddmmyy],
                  [yymmdd]

[NOAUTO], [STDLABEL], [volume id], [FIXED], [record length],
[AUTO]    (SL)          (F)
          (NL)          (FB)
          (NS)          (V)
          (BYPASS)
          (BLP)

[block length], [REWIND], [sequence number]
               (LEAVE)
               (UNLOAD)
    
```

59020267-0

**label1** This specifies the label of the file to be copied. The label for a disk file cannot exceed 8 characters. The label for a tape file cannot exceed 17 characters.

**Note:** This parameter is required for a standard label tape. An error message is sent if you specify label1 for a tape that is not a standard label tape.

**mmddyy, ddmmyy, or yymmdd**

This specifies the creation date of the file. If specified for the second parameter, it refers to the creation date of the file being copied. If specified for the seventh parameter, it refers to the creation date of an existing file and is valid only if ADD or ADDNOCHK is specified. The date must be specified in the same format as the session date. If you do not specify a creation date, and more than one file exists with the specified label, either the most recent date is used (if the file is a disk file) or the first file found is used (if the file is a tape file).

**F1** This specifies a disk file. If specified for the third parameter, F1 indicates that a disk file is being copied or added to a tape file. If specified for the sixth parameter, F1 indicates that a tape file is being copied or added to a disk file. If you specify T1 or T2 for the third parameter, F1 is assumed for the sixth parameter.

## **T1 and T2**

These specify tape files. If specified for the third parameter, T1 or T2 indicates that a tape file is being copied or added to a disk file. If specified for the sixth parameter, T1 or T2 indicates that a disk file is being copied or added to a tape file. If you specify F1 for the third parameter, T1 is assumed for the sixth parameter. T1 indicates that processing should begin with the tape mounted on tape drive 1. T2 indicates that processing should begin with the tape mounted on tape drive 2. TC is not allowed with the TAPECOPY procedure.

**ADD** This specifies that the file specified in the first parameter is to be added to the file specified in the fifth parameter. If a tape file is being added to, a check is made to ensure that the tape file is the last file on the tape. If the tape file is not the last file, an error message is sent. This error allows you to continue, so you can add to the tape file. However, files after the file to which you are adding are lost.

## **ADDNOCHK**

This specifies that the file specified in the first parameter is added to an existing file specified in the fifth parameter. ADDNOCHK is valid only when adding to a tape file. No check is made to see that it is the last file on the tape. Any files after the file to which you are adding are lost.

**NOADD** This specifies that the file specified in the first parameter is used to create a new output file. If you omit this parameter, NOADD is assumed.

**label2** This specifies the label of the file. If you specify ADD or ADDNOCHK, this parameter refers to the label of an existing disk or tape file to which records are added. If you specify NOADD, this parameter refers to the label of the new disk or tape file being created. If the file is a disk file, the label cannot exceed 8 characters. If the file is a tape file, the label cannot exceed 17 characters. If the label specified exceeds these limits, it is truncated. If this parameter is omitted, the label specified in parameter one (label1) is assumed.

## **retention days**

This specifies the retention period for the newly created tape file. Values of 0 to 999 are valid. If you do not specify this parameter, one day is assumed. If you specify a retention period of 999, the file becomes a permanent file. This parameter is valid only for tape files.

**NOAUTO** This specifies the tape drive (T1 or T2) to use. If you specify T1 or T2, only that specified tape drive is used for all tape volumes.

**AUTO** This specifies T1 or T2, so the system switches to the other tape drive (if available) to continue processing (T1 to T2 or T2 to T1) when the tape reel on the original tape drive is finished. If the other drive is not available, the original drive is used. A prompt is issued to verify that the mounted reel is the correct reel on which to continue processing.

If you do not specify a parameter, AUTO is assumed.

## **STDLABEL or SL**

This specifies that the tape to be processed is a standard label tape. If a tape file is to be copied or added to a disk file, or a disk file is to be added to an existing tape file, most of the information needed to process the tape can be taken from the tape file label. If a disk file is to be copied to a tape file, all information needed to process the file must be

supplied or taken from the disk file. If you omit this parameter, STDLABEL is assumed.

**NONLABEL or NL**

This specifies that the tape to be processed is a nonlabeled tape and all the information needed to process the file must be provided.

**NONSTAND or NS**

This specifies that the tape to be processed has nonstandard labels and all the information needed to process the file must be provided. NONSTAND is valid only when a tape file is being copied or added to a disk file.

**BYPASS or BLP**

This specifies that the tape has a standard tape label, but label processing is to be bypassed. All the information needed to process the file must be supplied. BYPASS is valid only when a tape file is being copied or added to a disk file.

**volume id**

This specifies the volume identification of the tape to be processed. The volume ID can be from 1 to 6 alphameric characters. It is used to check that the correct tape volume is mounted. If the correct tape volume is not mounted, an error message appears. The operator can do one of the following:

- Continue and process the file.
- Retry after mounting the correct tape.
- Cancel the procedure.

The volume ID parameter is required if a disk file is being copied to or added to a file on a standard label tape. This parameter is not allowed if you specify NONLABEL, NONSTAND, or BYPASS.

If a disk file is being copied or added to a tape file and the file requires more than one tape volume, the volume ID of the first volume is checked.

**FIXED or F**

This specifies that the record format of the file to be processed is fixed length, unblocked records. If you do not specify the record format, FIXED is assumed, except when reading or adding to a standard label file (REEL-SL) that has a HDR2 label. If the standard label file has a HDR2 label, the record format specified in the HDR2 label is used.

**FIXEDBLK or FB**

This specifies that the record format of the file to be processed is fixed length, blocked records.

**VARIABLE or V**

This specifies that the record format of the file to be processed is variable length, unblocked records. VARIABLE is valid only when a tape file is being copied or added to a disk file.

**Note:** If the type of processing is STDLABEL and the record format specified (or defaulted to) does not match the record format in the tape file label, an error message appears.

## record length

This specifies the number of bytes in a logical tape record. For variable length records, this is the maximum length. The value specified can be from 18 to 4096 bytes.

The record length parameter is required if you specify NONLABEL, NONSTAND, BYPASS, or STDLABEL and the file being processed does not have a HDR2 label. Also, the record length parameter is required if NONLABEL (NL) is specified and a disk file is being added to (not copied to) a tape file.

If you do not specify the record length when creating a new tape, the record length is the same as the disk file. If you specify the record length when creating a new tape file, and it is not the same as the disk file record length, an error message appears. The job can be canceled, or the job can be continued with the disk record padded or truncated to the record length specified.

If you do not specify the record length when a tape file is being copied or added to a disk file, or a disk file is being added to a tape file, the record length is taken from the tape file label. If you specify the record length, it must be the same as the record length in the tape file label.

## block length

This specifies the number of bytes in a physical block of data in the tape file. The value specified can be from 18 to 32 767 bytes. This parameter is required if you specify FIXEDBLK (FB) and NONLABEL (NL), NONSTAND (NS), or BYPASS (BLP). It is also required when a new tape file is created and the record format is FIXEDBLK (FB).

The block length is required if the tape is a STDLABEL (SL) tape that does not have an HDR2 label. The block length is optional when a tape file is being copied or added to a disk file or a disk file is being added to a tape file and you specify STDLABEL (SL). If you do not specify the block length, it is taken from the tape label. If you specify the block length the length must be the same as the block length in the tape file label.

The block length is not allowed if you specify FIXED (F) or VARIABLE (V). If you specify a block length greater than the region size, an error message appears.

**REWIND** This specifies that the tape is rewound after the TAPECOPY procedure has run. REWIND is assumed if you do not specify this parameter.

**LEAVE** This specifies that the tape is left where it is after the TAPECOPY procedure has run. The next step (within a procedure accessing the same tape unit) starts at this position. LEAVE information is maintained by the system from job step to job step, but is *not* passed from job to job.

**UNLOAD** This specifies that the tape is rewound and unloaded after the TAPECOPY procedure has run.

## sequence number

This specifies which file on the tape to process by its position on the tape. If this parameter is omitted, and you specify NONLABEL, NONSTAND, or BYPASS, a sequence number of 1 is assumed. If this parameter is omitted, and you specify STDLABEL, the file is found by its file label. If you specify a sequence number, and specify STDLABEL,

the file is located by sequence number first, then checked by the file label. If the file found by sequence number is not correct, the correct file is found by its file label.

**RECORDS,value1**

This specifies that the disk file being created must be large enough to contain the number of records specified by value1. You can specify a value from 1 to 8000000. If a tape file is being copied to a new disk file, either a RECORDS,value1 or a BLOCKS,value2 is required.

**BLOCKS,value2**

This specifies that the disk file being created must be large enough to contain the number of blocks specified by value2. You can specify a value from 1 to 312815. If a tape file is being copied to a new disk file, either a RECORDS,value1 or a BLOCKS,value2 is required.

**key length**

This specifies the key length of the indexed file that is being created. The value specified can be any decimal number from 1 to 120. You must specify the key length with the key location, and the sum of key length and key location cannot exceed the record length plus one. This parameter is not allowed when you create or add to a tape file or add to a disk file.

**key location**

This specifies the relative displacement of the start position of the record key for an indexed file being created. You must specify the key location with the key length. The sum of the key location and key length cannot exceed the record length plus one. This parameter is not allowed when you create or add to a tape file or add to a disk file.

**DUPKEY** This specifies that duplicate keys are allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If you do not specify this parameter when you create an indexed disk file, NODUPKEY is assumed.

**NODUPKEY**

This specifies that duplicate keys are not allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If you do not specify this parameter when you create an indexed disk file, NODUPKEY is assumed.

**Example 1**

This example copies a disk file called FILE1 to tape. The name of the tape file is FILE2. The tape is processed as a standard label tape with a volume ID of VOL001 and is mounted on tape drive 1. The tape is rewound after copying the file.

```
TAPECOPY FILE1,,F1,NOADD,FILE2,T1,,,STDLABEL,VOL001
```

**Example 2**

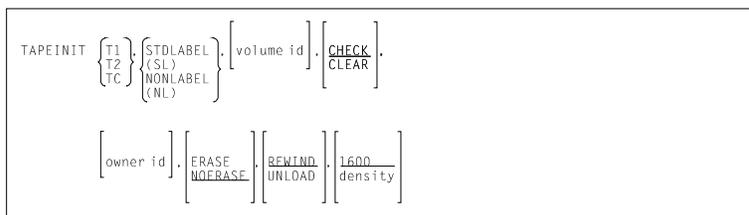
This example copies a file from a tape that contains 500 records to a disk file. The name of the tape file is FILE2, and the file to be created on disk is called FILE1. The tape is a standard label tape with a volume ID of VOL001 mounted on tape drive 1. The tape is rewound after copying the file.

```
TAPECOPY FILE2,,T1,,FILE1,F1,,,VOL001,,,,,RECORDS,500
```

## TAPEINIT Procedure

The TAPEINIT procedure prepares a tape so it can be used to save files and libraries. This preparation is called **initialization**. You can also use this procedure to erase a tape.

The TAPEINIT procedure runs the \$TINIT utility program.



RV3F061-0

- T1** This specifies that the tape to be prepared is on tape drive 1.
- T2** This specifies that the tape to be prepared is on tape drive 2.
- TC** This specifies that the tape to be prepared is a tape cartridge. STDLABEL (SL) is the only tape format allowed if you specify TC.

### STDLABEL or SL

This specifies that a labeled tape is prepared.

### NONLABEL or NL

This specifies that a nonlabeled tape is prepared.

**volume id** This specifies the volume identification written on the tape. You can specify up to 6 alphanumeric characters. This parameter is required if you specify STDLABEL (SL) and is not allowed if you specify NONLABEL (NL).

**CLEAR** This specifies that a new volume label is written without checking for an expired file.

**CHECK** This specifies that the first data file is checked to see if it is an expired file. If it is, a new volume label is written. If it is not an expired file, an error message appears. Do not specify CHECK for blank tapes. If you do not specify this parameter, CHECK is assumed.

**owner id** This specifies an additional identification field. You can specify up to 14 characters. If you do not specify an owner ID, the field is left blank. The owner ID is not allowed if you specify NONLABEL.

**ERASE** This specifies that after the new volume label is written, blanks are written to the end of the tape.

**NOERASE** This specifies that blanks are not written to the end of the tape after the new volume label has been written. If you do not specify this parameter, NOERASE is assumed. This parameter is not valid if you specify TC.

**REWIND** This specifies, for reel-to-reel tape, that the tape rewinds to the load point after processing completes.

For a tape cartridge, REWIND specifies that the tape is positioned to the beginning of the cartridge after processing completes.

- UNLOAD** This specifies, for reel-to-reel tape, that the tape rewinds and unloads after processing completes.
- For a tape cartridge, UNLOAD specifies that the tape is positioned to the end of the cartridge after processing completes.
- REWIND, LEAVE, and UNLOAD are valid only if the unit is tape (T1, T2, or TC).
- Each time a new or different tape cartridge is processed, or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The cartridge is wound to the end of the tape, then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, this process is lengthy. You can shorten the preparation time by specifying UNLOAD, because the tape is left at the end of the tape after the last operation completes.
- density** This specifies the recording format of the data to be written on the tape. If you do not specify a value, the density defaults to 1600. Any value supported for the DENSITY parameter of the INZTAP CL command can be specified.

### Example

This example initializes a tape on tape drive 1. The tape is initialized as a standard label tape with a volume ID of VOL001. The tape is checked to ensure that the first data file is an expired file. The tape is erased and then rewound and unloaded. The tape is initialized with a density of 38000.

```
TAPEINIT T1,SL,VOL001,,ERASE,UNLOAD,38000
```

---

## TAPESTAT Procedure

Not supported. On the System/36, the TAPESTAT procedure was used to display or print information about the tape volumes that were recorded in the volume statistical logs. If you were having trouble reading from or writing to a specific tape, the TAPESTAT procedure helped determine the type of errors the tape was producing.

To do a similar function, use the Print Error Log (PRERRLOG) command.

See the *CL Reference* book for more information on the PRERRLOG command. See the *System Operation* book for more information on working with system problems.

---

## TEXTCONV Procedure

Not supported. On System/36, the TEXTCONV procedure started the Text Conversion Aid so documents created with the Text Management System (TMS) could be converted to a form usable by DW/36.

On the AS/400 system, this function is not required. Documents created by TMS must be converted before being migrated to the AS/400 system.

See the book, *Using OfficeVision/400 Word Processing* for more information on the OfficeVision for OS/400 word processing functions. See the book, *Planning for and Setting Up OfficeVision/400* for more information on the folders and documents.

---

## TEXTDCT Procedure

The TEXTDCT procedure allows you to maintain a supplemental dictionary. For more information, see the *Using OfficeVision/400 Word Processing* book,

```
TEXTDCT
```

59020270-0

The TEXTDCT procedure has no parameters.

---

## TEXTDOC Procedure

The TEXTDOC procedure allows you to create or maintain a document.

For more information, see the *Using OfficeVision/400 Word Processing* book.

If you enter TEXTDOC with no parameters, the Work with Documents in Folders display appears.

To create, delete, adjust page endings, edit, or browse a document:

```
TEXTDOC [ CREATE
         DELETE
         PAGINATE
         REVISE
         VIEW ] , [ document name ] , [ folder name ] , [ subdirectory ]
```

59020271-1

To copy a document to another document:

```
TEXTDOC COPY , [ document name ] , [ folder name ] , [ new document name
               document name ] ,
               [ new folder name
               folder name ] , [ NOREPLACE
                               REPLACE ] , [ subdirectory ] , [ new subdirectory ]
```

59020272-2

To merge data from other documents or files into your document:

```
TEXTDOC MERGE , [ document name ] , [ folder name ] , [ new document name
               document name ] ,
               [ new folder name
               folder name ] , [ NOREPLACE
                               REPLACE ] , [ NOOPTIONS
                                             OPTIONS ] , [ subdirectory ] ,
               [ new subdirectory ]
```

59020527-2

To print a document:

```
TEXTDOC PRINT, [document name], [folder name], [NOOPTIONS], [subdirectory]
              [OPTIONS]
```

59020273-1

To rename a document:

```
TEXTDOC RENAME, [document name], [folder name], [new document name],
                [subdirectory]
```

59020274-1

To print documents to a disk file:

```
TEXTDOC PRTFILE, [document name], [folder name], [CHECK], [select status],
                [ALL], [NOCHECK]
                [new status], [TEXT], [filename], [ERRLOG], [DELETE], [subdirectory]
                [NOTEXT], [NOERRLOG], [NODELETE]
```

59020528-2

To check the spelling in a document:

```
TEXTDOC SPELL, [document name], [folder name], [beginning page], [ending page],
               [subdirectory]
```

59020558-1

**CREATE** This specifies that you want to create a new document.

**DELETE** This specifies that you want to remove a document from a folder.

**PAGINATE**

This specifies that you want to adjust the page endings for a document.

**REVISE** This specifies that you want to edit a document.

**VIEW** This specifies that you want to browse a document but not edit it.

**COPY** This specifies that you want to copy a document to another document. You can copy to the same folder (document) if you specify a different document (folder) name.

**MERGE** This specifies that you want to merge data or files into a document which can be edited.

**PRINT** This specifies that you want to print a document.

**RENAME** This specifies that you want to give an existing document a new name.

**PRTFILE** This specifies that you want to print one or more documents to a disk file.

**SPELL** This specifies that you want to check the spelling in your document. This is done as a batch job.

**document name**

This specifies the document you want to work with.

If you do not specify this parameter, the document you last worked with is used. If, however, you use the TEXTDOC CREATE or TEXTDOC DELETE procedures, you must specify this parameter.

If you specify PRTFILE, *document name* specifies the name of the document you want to have as output to a file. If you do not specify a document name, the Print Document prompt display appears.

**ALL** This specifies that you want to process all text documents within a folder or all text documents with a given status value within a folder. ALL is only valid if you specify PRTFILE.

**folder name**

This specifies the folder in which the document is stored or will be stored. If you specify PRTFILE, *folder name* specifies the name of the folder which contains the documents you want to process. If you do not specify this parameter, the folder you specified last is used.

**new document name**

This specifies the name of the new document. A document name can be up to 12 characters long and must begin with an alphabetic character (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). Avoid using the following characters, because they have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), equal signs (=), and hyphens (-).

**subdirectory**

This specifies the name of the subdirectory used. The subdirectory is stored in your profile and defaults to that profile if you do not specify it. If you have previously used a subdirectory, but now want to work out of the root directory, you must specify the root directory for the *folder* prompt, and leave the *subdirectory* prompt blank to override the profile value. A subdirectory name has the same restrictions as a folder name, except subdirectory names can have 12 characters (8-character name plus a 3-character extension).

**beginning page**

This specifies the page on which you want the spelling check to begin. This parameter is valid only if you specify SPELL.

**ending page**

This specifies the page on which you want the spelling check to end. This parameter is valid only if you specify SPELL.

**NOOPTIONS**

This specifies that the default print options are used for the document to be printed. If you do not specify a parameter, NOOPTIONS is assumed. If the first parameter is MERGE, NOOPTIONS specifies that the merge takes place with the default MERGE options. This parameter is valid only if you specify MERGE or PRINT.

**OPTIONS**

This specifies that the print options are displayed before the document is printed. If you change the print options when they are displayed, the

changes affect only the printing of this document. The changes are not saved. If the first parameter is MERGE and you specify OPTIONS, the Merge Options display appears, and you can specify options that affect how the information is merged into the new document. This parameter is valid only if you specify MERGE or PRINT.

**new folder name**

This specifies the name of the folder where the copied document is stored. If you are giving the new document the same name as the previous document, you must specify a name different from the previous folder. If you do not specify this parameter, *Folder name* is used.

A folder name has the same restrictions as a document name, except that the maximum folder name is 8 characters.

**NOREPLACE**

This specifies that if a document with the same name already exists, an error message appears. If you do not specify a parameter, NOREPLACE is assumed. This parameter is valid only if you specify MERGE or COPY.

**REPLACE**

This specifies that if a document with the same name already exists, the document is replaced. This parameter is valid only if you specify MERGE or COPY.

**CHECK**

This specifies that the document status is checked. This parameter is only valid if you specify PRTFILE. If you do not specify this parameter, the prompt defaults to the value you last used.

**NOCHECK**

This specifies that you do not want to make selections based on the document status. This parameter is only valid if you specify PRTFILE. If you do not specify this parameter, the prompt defaults to the value you last used.

**select status**

This specifies the status field of the text documents you want to select. This parameter is only valid if you specify CHECK with the PRTFILE option. If you do not specify this parameter, the prompt defaults to the value you last used.

**new status**

This specifies the new value for the status field of the documents selected. If you do not want the status to be changed, specify the same value for this parameter as for the select status parameter. This parameter is only valid if you specify CHECK with the PRTFILE option. If you do not specify this parameter, the prompt defaults to the value you last used.

**new subdirectory**

This specifies the name of the new subdirectory to be used. This name has the same restrictions as the subdirectory name.

**TEXT**

This specifies that you want the document text included with the document information in the file. This parameter is valid only if you specify PRTFILE. If you do not specify this parameter, the prompt defaults to the value you last used.

**NOTEXT** This specifies that you want only the document description records. This parameter is valid only if you specify PRTFILE. If you do not specify this parameter, it defaults to the value you last used.

**File name**

This specifies the name of the disk file to which you want to print your document. This parameter is valid only if you specify PRTFILE. If you do not specify this parameter, it defaults to the value you last used.

**ERRLOG** This specifies that you want the document error log included with the document. This allows you to see if any problems were encountered while the output disk file was created. This parameter is valid only if you specify PRTFILE. If you do not specify this parameter, it defaults to the value you last used.

**NOERRLOG**

This specifies that you do not want the document error log included in the document. This parameter is valid only if you specify PRTFILE. If you do not specify this parameter, it defaults to the value you last used.

**DELETE** This specifies that you want the document deleted after the disk file output is created. This parameter is valid only if you specify PRTFILE. If you do not specify this parameter, the prompt defaults to the value you last used.

**NODELETE**

This specifies that you want the document to remain in the folder. This parameter is valid only if you specify PRTFILE. If you do not specify this parameter, NODELETE is assumed.

**Note:** The sum of the characters in the folder name and subdirectory name (including forward slashes) cannot exceed 63 characters.

## File Format for PRTFILE Files

For each document copied to a disk file, one header record is created as the first record in that file. All records in PRTFILE files have a record length of 256 bytes.

**Header Record:** The header record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	1	B	X'20'
2	2	C	PF
4	253	C	Unused (unused positions are set to blanks)

Following the header record are the document records (record identifiers 10 to 99). A start document record is the first record for each document printed to the file.

**Start Document Record:** The start document record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (10)
3	4	C	Form (PF = print file)
7	12	C	Document name
19	8	C	Folder name (first 8 bytes if longer than 8)
27	62	C	Path
89	12	C	Full 12-character folder name
101	156	C	Unused (unused positions are set to blanks)

**Document Description Records:** Following the document records are document description records. The first document description record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (11)
3	60	C	Subject
63	35	C	Author
98	35	C	Receiver
133	1	C	Online (1 = yes, 2 = no)
134	16	C	Class
150	8	C	Date complete
158	8	C	Last revised date
166	8	C	Creation date
174	8	C	Action due date
182	8	C	Retention date
190	10	C	Project
200	10	C	Reference (first 10 bytes if longer than 10)
210	16	C	Keyword
226	1	C	Status (first byte if longer than 1)
227	2	B	Internal status (X'8000' = print as labels, X'4000' = object (graphics), X'0000' = other)
229	6	C	Last revised time
235	22	C	Unused (unused positions are set to blanks)

The second document description record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (12)
3	35	C	Description

## TEXTDOC

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
38	60	C	Capitalized subject
98	13	C	Original system
111	44	C	Original name
155	4	C	IDP character set/code page
159	4	C	Document character set/code page
163	1	C	Unused
164	1	C	Unused
165	2	B	Document type
167	17	C	Unused (unused positions are set to blanks)
184	8	C	Unused
192	8	C	Unused
200	6	C	Unused
206	1	C	Unused
207	2	B	Type available Bit 16 = AS/400 unresolved Bit 15 = AS/400 resolved Bit 14 = L2-DCA (FFT) Bit 13 = L3-DCA (RFT) Bit 12 = PC data Bit 11 = DW 4 internal Bit 10 = System/38 internal
209	1	C	PC mark for archive (1 = yes, 2 = no)
210	1	C	PC read only (1 = yes, 2 = no)
211	1	C	Unused
212	1	C	Unused
213	44	C	Unused (unused positions are set to blanks)

The third document description record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (13)
3	16	C	Owner
19	8	C	Date filed
27	4	C	Time filed (hours and minutes)
31	8	C	User associated document date
39	60	C	Full reference
99	20	C	Full status
119	60	C	Related document on source

<b>Beginning Column</b>	<b>Field Length</b>	<b>Character (C) or Binary (B)</b>	<b>Contents or Description</b>
179	1	C	Archive on next save 0=Not marked 1=Marked and save all 2=Marked and delete text 3=Marked and delete all
180	1	C	Read only document 0=Read only 1=Work in process
181	72	C	Unused (unused positions are set to blanks)

The fourth document description record contains the fields shown in the following table. This record may contain from 1 to 4 subjects.

<b>Beginning Column</b>	<b>Field Length</b>	<b>Character (C) or Binary (B)</b>	<b>Contents or Description</b>
1	2	C	Record identifier (14)
3	60	C	Capitalized subject
63	60	C	Capitalized subject
123	60	C	Capitalized subject
183	60	C	Capitalized subject
243	14	C	Unused (unused positions are set to blanks)

## TEXTDOC

The fifth document description record contains the fields shown in the following table. This record may contain up to 12 authors.

---

<b>Beginning Column</b>	<b>Field Length</b>	<b>Character (C) or Binary (B)</b>	<b>Contents or Description</b>
1	2	C	Record identifier (14)
3	20	C	Author
23	20	C	Author
43	20	C	Author
63	20	C	Author
83	20	C	Author
103	20	C	Author
123	20	C	Author
143	20	C	Author
163	20	C	Author
183	20	C	Author
203	20	C	Author
223	20	C	Author
243	14	C	Unused (unused positions are set to blanks)

---

The sixth document description record contains the fields shown in the following table. This record may contain from 1 to 7 keywords.

---

<b>Beginning Column</b>	<b>Field Length</b>	<b>Character (C) or Binary (B)</b>	<b>Contents or Description</b>
1	2	C	Record identifier (14)
3	35	C	Keyword
38	35	C	Keyword
73	35	C	Keyword
108	35	C	Keyword
143	35	C	Keyword
178	35	C	Keyword
213	35	C	Keyword
248	9	C	Unused (unused positions are set to blanks)

---

The seventh document description record contains the fields shown in the following table. This record may contain from 1 to 4 recipients.

---

<b>Beginning Column</b>	<b>Field Length</b>	<b>Character (C) or Binary (B)</b>	<b>Contents or Description</b>
1	2	C	Record identifier (14)
3	60	C	Recipients
63	60	C	Recipients
123	60	C	Recipients
183	60	C	Recipients
243	14	C	Unused (unused positions are set to blanks)

---

**General Format Record:** The general format record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (20)
3, 5, 7	2	B	Dictionaries 01 = American English 02 = UK English 03 = German 04 = Dutch 05 = National French 06 = Canadian French 07 = Italian 08 = Spanish 09 = Swedish 10 = Finnish 11 = Danish 12 = Norwegian 13 = Portuguese 14 = Icelandic 15 = Greek 16 = Afrikaan 17 = Swiss-German 19 = Catalan 51 = American English Legal 76 = American English Medical
9	4	B	System text unit name
13	2	B	Text unit name processing
15	2	B	Right-to-left data exists (0 = no, 1 = yes)
17	168	C	Month names (array of 12 names, each 14 characters long)
185	1	C	Change symbol character
186	4	C	Print forms number
190	8	C	File/query name (first 8 characters if longer than 8)
198	8	C	Query library (first 8 characters if longer than 8)
206	2	B	Graphic character set ID
208	2	B	Code page global ID
210	10	C	Full file/query name
220	10	C	Full query library name
230	27	C	Unused (unused positions are set to blanks)

**Page Format Record:** The page format record contains the fields shown in the following table:

Table 4-3 (Page 1 of 2). Page Format Record Fields

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (30)
3	2	B	Format ID number

Table 4-3 (Page 2 of 2). Page Format Record Fields

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
5	8	C	Format name
13	40	C	Description
53	4	B	Page width
57	4	B	Page depth
61	2	B	First typing line for first page
63	2	B	First typing line for subsequent pages
65	2	B	Last typing line
67	2	B	First page forms type (0 = current, 1 = paper, 2 = envelope)
69	2	B	Other pages forms type (0 = current, 1 = paper, 2 = envelope)
71	2	B	First page paper source (0 = current, # = drawer number)
73	2	B	Other pages paper source (0 = current, # = drawer number)
75	2	B	Paper feed
77	2	B	Offset stack
79	2	B	Destination drawer
81	2	B	Quality of print
83	2	B	Duplex print
85	1	C	Header all pages (1 = yes, 2 = no)
86	1	C	Header even pages (1 = yes, 2 = no)
87	1	C	Header odd pages (1 = yes, 2 = no)
88	1	C	Header on first page (1 = yes, 2 = no)
89	1	C	Footer on all pages (1 = yes, 2 = no)
90	1	C	Footer on even pages (1 = yes, 2 = no)
91	1	C	Footer on odd pages (1 = yes, 2 = no)
92	1	C	Footer on first page (1 = yes, 2 = no)
93	2	B	First line for header
95	2	B	First line for footer
97	2	B	Odd header length
99	2	B	Even header length
101	3	C	Degrees to rotate, blanks imply auto
104	153	C	Unused (unused positions are set to blanks)

**Line Format Record:** The line format record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (40)

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
3	2	B	Format ID number
5	8	C	Format name
13	40	C	Description
53	1	C	Adjust (1 = yes, 2 = no)
54	1	C	Print half index up (superscript)/half index down (subscript)/word underscore as spaces (1 = yes, 2 = no)
55	1	C	Print underline/bold as spaces (1 = yes, 2 = no)
56	1	C	Auto hyphen (1 = yes, 2 = no)
57	1	C	Compress wide paragraphs (1 = yes, 2 = no)
58	2	B	Type style number (font ID)
60	2	B	Lines per inch
62	2	B	Document left-most left margin
64	2	B	Left margin
66	2	B	Right margin
68	2	B	Spacing 1 = one half 2 = single 3 = one and one half 4 = double 5 = two and one half 6 = triple
70	2	B	Zone width
72	2	B	Line alignment (justification percent)
74	2	B	Number of tabs (0 through 48)
76	96	B	Array of tab positions (2 bytes per tab)
172	48	C	Array of tab types (1 character per tab) 3 = left 4 = right 5 = center 6 = decimal 7 = comma 8 = colon
220	2	B	Line numbering state 1 = on, resume on next line 2 = on, reset on next line 3 = off, continue numbering 4 = off, suspend numbering
222	2	B	Line orientation 0 = left-to-right 1 = right-to-left

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
224	2	B	Color 0 = base 1 = blue 2 = red 3 = pink 4 = green 5 = turquoise 6 = yellow 8 = black 16 = brown
226	2	B	Font width (in 1440ths of an inch)
228	2	B	Font spacing 0 = based on font ID 1 = mono 2 = proportional
230	2	B	Graphic character set ID
232	2	B	Code page global ID
234	23	C	Unused (unused positions are set to blanks)

### Text Lines

The following records define text lines (record identifiers 50 to 79). If there are more than 235 bytes of text in a line, a second record is created with the start position of 236. The second record has the required carrier return indicator set to 1 for *Yes* or 2 for *No*, but the required carrier return indicator for the first record is blank (indicating that the line did not end on this record).

**Text Half Index Up (Superscript) Record:** The following table defines text lines for the text half index up (superscript) record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (50)
3	2	B	Page number
5	2	B	Line number
7	2	B	Print position of first character in line (1, 236, etc.)
9	1	C	Margin text (1 = yes, 2 = no)
10	2	B	Position of last nonblank character in line
12	1	C	Required carrier return (1 = yes, 2 = no)
13	9	C	Unused (unused positions are set to blanks)
22	235	C	Line

**Text Bold Record:** The following table defines text lines for the text bold record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (51) <sup>1</sup>
1 The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

***Text Half Index Up (Superscript) Underline Record:*** The following table defines text lines for the text half index up (superscript) underline record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (52) <sup>1</sup>
1 The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

***Text Half Index Up (Superscript) Overstrike Record:*** The following table defines text lines for the text half index up (superscript) overstrike record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (53) <sup>1</sup>
1 The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

***Text Half Index Up (Superscript) Backspace Record:*** The following table defines text lines for the text half index up (superscript) backspace record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (54) <sup>1</sup>
1 The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

***Text Base Line Record:*** The following table defines text lines for the text base line record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (60) <sup>1</sup>
1 The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

***Text Base Bold Record:*** The following table defines text lines for the text base bold record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (61) <sup>1</sup>
1 The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Text Base Underline Record:** The following table defines text lines for the text base underline record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (62) <sup>1</sup>
<sup>1</sup> The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Text Base Overstrike Record:** The following table defines text lines for the text base overstrike record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (63) <sup>1</sup>
<sup>1</sup> The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Text Base Backspace Record:** The following table defines text lines for the text base backspace record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (64) <sup>1</sup>
<sup>1</sup> The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Text Half Index Down (Subscript) Record:** The following table defines text lines for the text half index down (subscript) record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (70) <sup>1</sup>
<sup>1</sup> The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Text Half Index Down (Subscript) Bold Record:** The following table defines text lines for the text half index down (subscript) bold record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (71) <sup>1</sup>
<sup>1</sup> The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Text Half Index Down (Subscript) Underline Record:** The following table defines text lines for the text half index down (subscript) underline record:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (72) <sup>1</sup>
<sup>1</sup> The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Text Half Index Down (Subscript) Overstrike Record:** The following table defines text lines for the text half index down (subscript) overstrike record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (73) <sup>1</sup>
<sup>1</sup> The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Text Half Index Down (Subscript) Backspace Record:** The following table defines text lines for the text half index down (subscript) backspace record.

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (74) <sup>1</sup>
<sup>1</sup> The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record" on page 4-276.			

**Instruction Record:** The instruction record is used for the Change Font control, the Set Color control, and any unrecognized or pass-through controls. The instruction record contains the fields shown in the following table:

Table 4-4. Instruction Record Fields

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (80)
3	2	B	Page number
5	2	B	Line number
7	1	C	Continuation record follows (1 = yes, 2 = no)
8	2	B	Position of instruction
10	2	B	Length of instruction
12	10	C	Unused (unused positions are set to blanks)
22	235	C	Instruction (may contain values less than '40'X)

**Start Error Log Record:** The start error log record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (90)
3	4	C	Form
7	12	C	Document
19	8	C	Folder
27	62	C	Subdirectory
89	12	C	Full 12-character folder name

## TEXTLOAD

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
101	156	C	Unused (unused positions are set to blanks)

**End Document Record:** The end document record contains the fields shown in the following table:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (99)
3	4	C	Form
7	12	C	Document
19	8	C	Folder
27	62	C	Subdirectory
89	12	C	Full 12-character folder name
101	156	C	Unused (unused positions are set to blanks)

## TEXTFLDR Procedure

The TEXTFLDR procedure allows you to create or maintain a folder. When you enter TEXTFLDR, the Work with Folders display is shown.

See the book, *Using OfficeVision/400 Word Processing*, for more information.

```
TEXTFLDR [ folder name ]
```

59020275-0

### folder name

This parameter is ignored. Use the Work with Folders display to create folders. A folder name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, \$, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). Avoid using the following characters, because they have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), equal signs (=), and hyphens (-).

## TEXTLOAD Procedure

Not supported. On System/36, the TEXTLOAD procedure installed the DW/36 support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## TEXTOBJ Procedure

The TEXTOBJ procedure allows you to maintain nontext document data. Use the Work with Nontext Document Data display to list the names of all the nontext document data in the folder you specify. You can also select an option to copy, delete, or rename nontext document data, such as graphs and images that are stored in a document.

TEXTOBJ	[ folder name ]
---------	-----------------

59020593-0

### folder name

This specifies the folder that contains the nontext document data with which you want to work. If you do not specify this parameter, the procedure uses the name of the folder you last used.

---

## TEXTPROF Procedure

The TEXTPROF procedure allows you to work with text profiles or the system profile. Use the Work with Text Profile display to create, copy, revise, delete, or activate a text profile. You can also use this display to revise, copy, or activate the system profile.

TEXTPROF
----------

59020619-0

The TEXTPROF procedure has no parameters.

---

## TEXTPRTQ Procedure

The TEXTPRTQ procedure allows you to work with documents that are in the print queue. Use the Work with Documents to be Printed display to change, hold, delete, or release documents to be printed.

For more information, see the book, *Using OfficeVision/400 Word Processing*.

TEXTPRTQ
----------

59020559-0

The TEXTPRTQ procedure has no parameters.

---

## TEXTREL Procedure

Not supported. On the System/36, the TEXTREL procedure released documents that had been held for later printing.

To do a similar function, use the Work with Document Print Queue (WRKDOCPRTQ) command or use the "TEXTPRTQ Procedure."

See the *CL Reference* book for more information on the WRKDOCPRTQ command. See the book, *Using OfficeVision/400 Word Processing*, for more information on the OfficeVision for OS/400 word processing functions.

## TEXTSAVE Procedure

Not supported. On the System/36, the TEXTSAVE procedure saved the DW/36 support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

## TIMER Procedure

The TIMER procedure allows a user with sufficient authority to indicate whether a system should power on and perform an unattended initial program load (IPL). Three types of unattended IPL operations are supported by the procedure:

- An unattended IPL after a power failure
- An unattended IPL from a remote site using a communications modem
- An unattended IPL at a specific time and date

To use the TIMER procedure to either enable or disable any of the three unattended IPL operations, a user must have both \*ALLOBJ and \*SECADM authority, as well as \*USE authority to the CHGSYSVAL CL command. In order for an unattended IPL to occur, the system must have either lost and subsequently regained power or presently be turned off. The keylock switch on the control panel must be set to either the **Normal** or the **Auto** position. Following an unattended IPL, the Sign On display appears.

The TIMER procedure also allows any user to determine whether the system has been previously prepared to perform any of the three unattended IPL operations.

The TIMER procedure processes a TIMERSET OCL statement. See "TIMERSET OCL Statement" on page 5-100 for more information.

```

TIMER      [ QUERY
            REMOTE
            NOREMOTE
            RESTART
            NORESTART
            TIMED
            NOTIMED ] , { mddy } , { ddmyy } , { hhmmss }
            , { yymmdd }
    
```

RV2P178-0

### QUERY

Specifies that the current system values for the three unattended IPL operations should be listed on the current system list device.

**Note:** The QUERY parameter cannot be used with any other parameter in the TIMER procedure.

### REMOTE

Specifies that the system should power on and perform an unattended IPL when a telephone call is received through a modem.

### NOREMOTE

Specifies that the system should not power on and perform an unattended IPL when a telephone call is received through a modem.

**RESTART**

Specifies that the system should power on and perform an unattended IPL after a power failure occurs.

**NORESTART**

Specifies that the system should not power on and perform an unattended IPL after a power failure occurs.

**TIMED**

Specifies that the system should power on and perform an unattended IPL at a specific time and date.

If TIMED is specified, a date and time must also be specified.

The system value set with the TIMED parameter is not cleared when the specified date and time are reached and the unattended IPL takes place. The system value remains as set until it is changed through another use of the TIMED parameter.

**NOTIMED**

Specifies that the system should not power on and perform an unattended IPL at a specific time and date.

**mmddy, ddmmy, yymmdd**

Specifies the month (mm), day (dd), and year (yy) that the system should power on and perform an unattended IPL. The month, day, and year must be specified in the session date format. If the date specified is a past date, an error message is issued. This parameter can only be specified when the TIMED parameter value is specified.

**hhmmss**

Specifies the hour (hh), minute (mm), and second (ss), in 24-hour format, that the system should power on and perform an unattended IPL. The value specified for seconds is ignored. The time must be at least five minutes greater than the current time; otherwise, an error message is sent. This parameter can only be specified when the TIMED parameter value is specified.

**Example**

This example shows how to prepare a system to perform an unattended power on and IPL at a specific time and date.

```
TIMER TIMED,121291,063000
```

---

## TOLIBR Procedure

The TOLIBR procedure copies into a library a disk, diskette, tape, or tape cartridge file that contains one or more library members. You can also copy selected members from the file to the library.

All sector mode files copied by TOLIBR must be created either by the FROMLIBR procedure or by the \$MAINT utility program on the AS/400 system.

If a sector mode disk file is copied to a library, the name of the library must be the same as the name of the library from which the members were copied.

Each library member in a record mode file copied by TOLIBR must begin with a COPY statement and end with a CEND statement. Refer to the "COPY and CEND

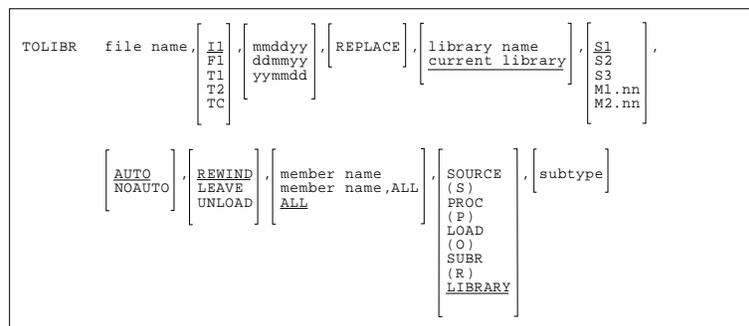
Statements” on page A-47 for the format of these statements. COPY and CEND statements are automatically inserted in members created by \$MAINT. You must insert them at the beginning and end of members not created by the \$MAINT utility program or by the FROMLIBR procedure. Do not insert more than the one required CEND statement. If a CEND statement exists within the member, an error message appears.

If the record mode file is organized as a direct disk file, you must insert an END statement following the last CEND statement in the file. The format of the END statement is as follows:

```
// END
```

where only one blank can separate the // and the END.

The TOLIBR procedure runs the \$MAINT utility program.



**file name** This specifies the file containing the one or more library members to be copied into the library.

**I1** This specifies that the file is on diskette. If you do not specify a parameter, I1 is assumed.

**F1** This specifies that the file is on disk.

**T1, T2, and TC** These specify that the file is on tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

**mmddyy, ddmmyy, and yymmdd** This specifies the creation date of the file containing the members to be copied. The date, if specified, must be in the session date format. Use the STATUS SESSION command to determine the session date format. If more than one file exists with the specified file name, and if the date is not specified, the following applies:

- If I1 is specified or assumed, the first file with that name on the diskette is copied.
- If you specified F1, the most recently created file with that name is copied.
- If you specify T1, T2, or TC, the first file with that name on tape is copied.

**REPLACE** This specifies that if library members already exist with the specified library name, they are to be replaced. If you specify REPLACE, new members replace existing members with duplicate

names, and no messages regarding the replacements are displayed.

In record mode, no message is issued on duplicates, and members are always replaced. In sector mode, if you do not specify REPLACE, members are placed in the library until a duplicate is found, at which time the system displays a message telling the operator that a duplicate exists. In response to the message, the operator can either cancel the job or continue processing. If the job is continued, the new member replaces the existing member in the library. If other duplicates are found during the job, existing members are automatically replaced and no messages are displayed regarding the duplicate members.

**library name** This specifies the library that is to contain the copied members. If you do not specify a library name, the current library is assumed. The library name cannot be QSYS, QSSP, or QSYS38. If a FROMLIBR was used to create a disk sector mode file, this must be the same library name that was used in the FROMLIBR procedure.

**S1, S2, S3, M1.nn, and M2.nn**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**AUTO** This specifies that when the tape reel (unit T1 or T2) on the original tape drive finishes, the system switches to the other tape drive to continue processing. If the tape reel on the other tape drive is not the same density as the original tape reel, an error message is sent. If the other tape drive is not available, the system uses the original tape drive.

**NOAUTO** This specifies that when the original tape drive finishes (unit T1, T2, or TC), the system prompts you to mount the next tape reel on the original tape drive. The next tape reel must have the same density as the first tape reel.

If the unit is tape, and neither AUTO nor NOAUTO is specified, AUTO is assumed.

If TC is specified for the unit, the AUTO/NOAUTO parameter is ignored.

If the unit is diskette (I1), the AUTO/NOAUTO parameter is supported for compatibility only. If specified, the value is syntax-checked, but is ignored when the diskette is processed.

**REWIND** This specifies that a reel-to-reel tape is rewound to the load point after processing is completed. It also specifies that a tape cartridge is positioned to the beginning of the cartridge after processing is completed.

**LEAVE** This specifies that the tape or tape cartridge is not rewound after the SAVE procedure runs. The next step within the same job that accesses the tape drives starts at this position if you specify the same tape unit.

- UNLOAD** This specifies that a reel-to-reel tape drive is rewound and unloaded after processing is completed. It also specifies that a tape cartridge is positioned to the end of the cartridge after processing is completed.
- REWIND, LEAVE, or UNLOAD are valid only if the unit is tape (T1, T2 or TC).
- Each time a new or different tape cartridge is processed, or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. To do this, the tape is wound to the end, and then rewound back to the beginning. If REWIND or LEAVE was specified for the last operation, this can be a lengthy process. The preparation time can be shortened if UNLOAD is specified, because the tape is left at the end after the last operation is completed.
- member name** This specifies the library member to be copied.
- member name,ALL**  
This specifies that the library members whose names begin with the specified characters (member name) are copied. You can specify up to 7 characters for member name.
- ALL** This specifies that all members of the library or within a certain subtype are copied. If you do not specify a parameter, ALL is assumed.
- SOURCE or S** This specifies that only library source members are copied. Source members are all the members in a source file named QS36SRC in the specified library.
- PROC or P** This specifies that only the library procedure members are copied. Procedure members are all the members in a source file named QS36PRC in the specified library.
- LOAD or O** This specifies that only the library load members are copied. Load members are all objects in a library with the following OS/400 object types:
- \*PGM Programs
  - \*MSGF Message files
  - \*FILE Display files (attribute of \*DSPF)
- SUBR or R** This specifies that only the library subroutine members are copied. Subroutine members are all objects in a library with the OS/400 object type of \*PGM (programs).
- LIBRARY** This specifies that all library member types (SOURCE, PROC, LOAD, and SUBR) are copied, including all objects with the following OS/400 object types:
- \*CHTFMT Chart formats
  - \*CLS Classes
  - \*CMD Commands
  - \*DTAARA Data areas
  - \*FCT Forms control tables
  - \*FILE Files (files with all attributes: PF, LF, PRTF, and so on)
  - \*GSS Graphic symbol sets

<b>*JOB</b>	Job descriptions
<b>*JOBQ</b>	Job queues
<b>*MENU</b>	Menus
<b>*MSGQ</b>	Message queues
<b>*OUTQ</b>	Output queues
<b>*QRYDFN</b>	Query definitions
<b>*SBSD</b>	Subsystem descriptions
<b>*SSND</b>	Session descriptions
<b>*TBL</b>	Tables

If you do not specify a parameter, LIBRARY is assumed.

**subtype** This specifies the subtype of the members to be copied. If you do not specify the subtype, it is not used as a qualifier when members are selected. Valid subtypes are as follows:

Subtype to Enter	Subtype Displayed	Meaning
ARP	RPT or RPT36	RPG auto report member
ARS	ARS36	Automatic response member
ASM	ASM36	Assembler member
BAP	BASP or BASP36	BASIC procedure (source member)
BAS	BAS or BAS36	BASIC member
BGC	BGC36	Business graphics chart
BGD	BGD36	Business graphics data
BGF	BGF36	Business graphics format
COB	CBL or CBL36	COBOL member
CSM		Communications and system management member <sup>1</sup>
CSP		Cross-system product member <sup>1</sup>
DFU	DFU36	Data file utility member
DLS		Document library service member <sup>1</sup>
DTA	DTA36	Data member
FMT	DSPF or DSPF36	Display format member
FOR	FOR36	FORTRAN member
ICF		CONFIGICF procedure member <sup>1</sup>
KEY		KEYS procedure member <sup>1</sup>
MNU	MNU or MNU36	Menu member
MSG	MSGF or MSGF36	Message member
PHL	PHL36	Phone list member
QDE		Query data entry member <sup>1</sup>
QRY		Query member <sup>1</sup>
QRY		Query member <sup>1</sup>
RPG	RPG or RPG36	RPG member
SRT	SRT36	Sort member
SSP		CNFIGSSP procedure member <sup>1</sup>

## TRACE

---

Subtype to Enter	Subtype Displayed	Meaning
TXT	TXT	Text member
UNS	UNS36	Unspecified
WSU	WSU36	Work station utility member
X25		X.25 packet switching control link <sup>1</sup>

---

<sup>1</sup> These subtypes are supported for compatibility only. No OS/400 subtype corresponds to these System/36 subtypes. If one of these subtypes is specified, no members are found.

---

### Example 1

This example copies the members from a diskette file called PAY into a library named MYLIB and replaces any duplicate members with no warning message:

```
TOLIBR PAY, ,,REPLACE,MYLIB
```

### Example 2

This example copies source members only from a diskette file named PAY into a library named MYLIB and replaces any duplicate members with no warning message:

```
TOLIBR PAY, ,,REPLACE,MYLIB, , , ,SOURCE
```

### Example 3

This example copies only the procedure member named XYZ from a diskette file named PAY into a library named MYLIB. It issues a message if a member with that name already exists and the file is a sector mode file. If the file is a record mode file, no message appears, but the member is replaced.

```
TOLIBR PAY, , , ,MYLIB, , , ,XYZ,PROC
```

### Example 4

This example copies only COBOL members from a diskette file named PAY into a library named MYLIB. It issues a message if a member with that name already exists and the file is a sector mode file. If the file is a record mode file, no message appears, but the member is replaced.

```
TOLIBR PAY, , , ,MYLIB, , , , ,COB
```

---

## TRACE Procedure

Not supported. On System/36, the TRACE procedure kept a history of events that occurred on the system. Selected system events were recorded as they occurred in a variable-length, wrap-around table in main storage.

To do a similar function, use the Trace Internal (TRCINT) command, or use the service functions provided by the Start System Service Tools (STRSST) command. The service functions allow you to trace the AS/400 vertical microcode, or allow you to work with a communications trace.

Use the Trace Job (TRCJOB) command to trace information for a particular job, or use the Start Service Job (STRSRVJOB) and End Service Job (ENDSRVJOB) commands to trace someone else's job.

See the *CL Reference* book for more information on the TRCINT, STRSST, TRCJOB, STRSRVJOB, and ENDSRVJOB command. See the *System Operation* book for more information on working with system problems. See the *AS/400 Licensed Internal Code Diagnostic Aids – Volume 1* book for more information on the system service tools (SST).

---

## TRANSFER Procedure

The TRANSFER procedure copies basic data exchange or I-exchange diskette files to a sequential or an indexed disk file. It also creates basic data exchange or I-exchange diskette files from a sequential, direct, or indexed disk file. See “Basic Data Exchange Files” on page 4-293 and “I-Exchange Files” on page 4-293 for more information about these types of files.

To display or print a basic data exchange or I-exchange diskette file, see the “LISTFILE Procedure” on page 4-142.

The TRANSFER procedure allows you to do the following:

- Convert a sequential, indexed, or direct disk file to a basic data exchange or I-exchange diskette file.
- Convert a basic data exchange or I-exchange diskette file to a sequential or indexed disk file.
- Add a basic data exchange or I-exchange diskette file to an existing sequential disk file.

Deleted records in a delete-capable file are not transferred to diskette.

When a basic data exchange or I-exchange diskette file is transferred to a sequential or indexed disk file, records are placed in the disk file sequentially, using the record length of the diskette file.

When a basic exchange diskette file is created from a disk file, the record length of the diskette file is set to that of the disk file, to 128 (for diskette 1), or to 256 (for diskette 2D), whichever is smaller. When an I-exchange diskette file is created from a disk file, the record length of the diskette file is set to that of the disk file.

When a basic data exchange diskette file is added to an existing sequential disk file, the record length of the disk file is used for all records added to the file. This causes the records in the diskette file to be either truncated or padded with zeros (X'00') if their record length is different from the existing disk file.

When an I-exchange diskette file is added to an existing sequential disk file, the record length of the diskette file must equal the record length of the disk file.

When you are creating a diskette file from a disk file, the diskette file name cannot contain a period. (Rename the disk file to a name which does not contain a period, before copying it to the diskette.)

The TRANSFER procedure runs the \$BICR utility program. The \$BICR utility program processes records sequentially. If the input file is an indexed file and the output file is a basic data exchange diskette file, the records are read sequentially by key. If the output file is an I-exchange diskette file, the records are read sequentially by the physical order in which they exist in the file.

# TRANSFER

To copy a basic data exchange or I-exchange diskette file to a new disk file:

```
TRANSFER input file name, [I], [mmddyy
ddmmyy
yymmdd], [NOADD], {key length,key position},
{RECORDS,records
BLOCKS,blocks}, [S1
S2
S3
M1.nn
M2.nn], [AUTO
NOAUTO], [NODUPKEY
DUPKEY]
```

59020280-0

To add a basic data exchange or I-exchange diskette file to an existing disk file:

```
TRANSFER input file name, [I], [mmddyy
ddmmyy
yymmdd], ADD, [output file name
input file name], [date],
[S1
S2
S3
M1.nn
M2.nn], [AUTO
NOAUTO]
```

59020281-0

To copy a disk file to a basic data exchange or I-exchange diskette file:

```
TRANSFER input file name,F1, [mmddyy
ddmmyy
yymmdd], volume id, [retention days],,,
[S1
S2
S3
M1.nn
M2.nn], [AUTO
NOAUTO], [EXCHANGE
IFORMAT]
```

59020282-0

## input file name

This specifies the file being transferred. If you are creating a new file, it is given the specified file name.

**I1** This specifies that a basic data exchange or I-exchange diskette file is being transferred to a sequential or indexed disk file. If you do not specify a parameter, I1 is assumed.

**F1** This specifies that a disk file is being transferred to a basic data exchange or I-exchange diskette file.

## mmddyy, ddmmyy, and yymmdd

These specify the creation date of the file being transferred. The date, if specified, must be in the session date format. Use the STATUS SESSION command to determine the session date format. If more than one file exists with the specified file name, and if the date is not specified, the following apply:

- If you specify I1, the first file with that name is transferred.
- If you specify F1, the most recently created file with that name is transferred.

**NOADD** This specifies that the basic data exchange or I-exchange diskette file being transferred becomes a new disk file. NOADD is assumed whenever a file is transferred from diskette to disk if ADD is not specified.

**ADD** This specifies that the input diskette file is added to the output disk file. (The first record in the input file immediately follows the last record in the output file.)

#### **key length**

This specifies the key length for the indexed disk file being created. The key length can be any number from 1 to 120. You must specify the number with the key position. The sum of the key length and the key position must not exceed the length of the record.

#### **key position**

This specifies the starting position of the key for the indexed disk file being created. For basic data exchange files, the key position can be any number from 1 to 128 for type 1 diskettes or from 1 to 256 for type 2D diskettes. For I-exchange files, the key location can be any number from 1 to 4096. You must specify the key location with the key length. The sum of the key length and the key position must not exceed the length of the record. If the key length and key position are not specified, a sequential file is created.

**DUPKEY** This specifies that duplicate keys are allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If you do not specify this parameter, the attribute of the input file is the attribute of the output file.

#### **NODUPKEY**

This specifies that duplicate keys are not allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If you do not specify this parameter, the attribute of the input file is the attribute of the output file.

#### **RECORDS**

This specifies that the disk file being created must be large enough to contain the number of records specified. It also specifies the number of records for the file. You can specify any number from 1 to 8 000 000.

Either RECORDS or BLOCKS is required under the following conditions:

- The file being transferred is on more than one diskette.
- The created disk file is larger than the file being transferred.

**BLOCKS** This specifies that the disk file being created must be large enough to contain the number of blocks specified. It also specifies the number of blocks for the file. Specify any number from 1 to 312 815.

#### **output file name**

This specifies the existing disk file to which a basic data exchange or I-exchange diskette file is added. The output file name is allowed only if ADD is specified. If you do not specify an output file name, the input file name is assumed.

**date** This specifies the creation date of the existing disk file. The date is allowed only if you specify ADD. You must specify the date in the same format as the session date. You can use the STATUS SESSION command to determine the format of the date.

## TRANSFER

### **volume id**

This specifies the volume ID of the diskette. You can specify from 1 to 6 alphanumeric characters.

### **retention days**

This specifies the length of the retention period (in days) for the created basic data exchange or I-exchange diskette file. You can specify any number from 1 through 999. If you do not specify a retention period, 1 day is assumed. If you specify a retention period of 999 days, the diskette file is a permanent file. For more information on diskette file retention, see the "FILE OCL Statement (for Diskette Files)" on page 5-33.

### **S1, S2, S3, M1.nn, and M2.nn**

These are supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

### **AUTO, NOAUTO**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

### **EXCHANGE**

This specifies that a disk file is transferred to a basic data exchange diskette file. If you do not specify a parameter, EXCHANGE is assumed.

### **IFORMAT**

This specifies that a disk file is transferred to an I-exchange diskette file.

### **Example 1**

This example creates a disk sequential file named FILE2 from a diskette basic data exchange or I-exchange file named FILE2:

```
TRANSFER FILE2
```

### **Example 2**

This example creates an indexed disk file named FILE2 from a basic data exchange file named FILE2. The key is in positions 1 to 5.

```
TRANSFER FILE2,,,,5,1
```

### **Example 3**

This example adds a basic data exchange or I-exchange diskette file named FILE1 to an existing disk file named FILE1:

```
TRANSFER FILE1,,,ADD
```

### **Example 4**

This example creates a basic data exchange diskette file named FILE3 on a diskette from a disk file named FILE3. The file is saved for 30 days. The volume ID of the diskette is VOL001.

```
TRANSFER FILE3,F1,,,VOL001,30
```

### Example 5

This example creates an I-exchange diskette file named FILE3 on a diskette from a disk file named FILE3. The file is saved for 30 days. The volume ID of the diskette is VOL002.

```
TRANSFER FILE3,F1,,VOL002,30,,,,IFORMAT
```

## Basic Data Exchange Files

When you create or copy basic data exchange diskette files, the diskettes being used must be initialized in one of the following formats, using the INIT procedure's FORMAT parameter:

- Diskette 1 (one-sided) diskettes must be initialized in the 128-bytes-per-sector format.
- Diskette 2D (two-sided, double-density) diskettes must be initialized in the 256-bytes-per-sector format.

If you do not know the diskette format, use the CATALOG procedure to list the diskette format.

These diskette files can be used to exchange data between systems and devices. See the *IBM Diskette General Information Manual* for information about the data set label fields.

The records in a basic data exchange file are not blocked and cannot span diskette sectors. Only one diskette sector (128 bytes for the diskette 1, 256 bytes for the diskette 2D) is used for each record. The data is truncated if the record length is greater than the diskette sector size.

## I-Exchange Files

You can use these diskette files to exchange data between systems and devices that support the diskette exchange type I.

When you copy I-exchange files, the diskettes must be initialized in one of the following formats, using the INIT procedure's FORMAT or FORMAT2 parameter:

- Diskette 1 (one-sided) can be initialized in either the 128- or 512-bytes-per-sector format.
- Diskette 2D (two-sided, double density) can be initialized in either the 256- or 1024-bytes-per-sector format.

The records in an I-exchange file are blocked and can span diskette sectors. Several records and parts of records can be placed in a diskette sector, or a record can extend from one sector to another. However, the records cannot span diskette volumes.

---

## TRNMGR Procedure

Not supported. On System/36, the TRNMGR procedure allowed you to start, stop, or change the error reporting level for a specified line in an IBM Token-Ring Network. It provided several reporting levels so that you could control the amount of information logged in the system history file.

## UPDATE

To do a similar function, use the Create Line Description for Token-Ring (CRTLINTRN), Change Line Description for Token-Ring (CHGLINTRN), and Vary Configuration (VRYCFG) commands. The error-reporting level for the token-ring network is set by specifying a value for the TRLAN manager-logging-level (TRNLOGLVL) parameter on the CRTLINTRN or CHGLINTRN commands. Error reporting is available when the line is varied on (use the VRYCFG command), based on the current logging level. The reporting level can be changed at any time, using the CHGLINTRN command.

See the *CL Reference* book for more information on the VRYCFG, CRTLINTRN, and CHGLINTRN commands. See the *System/36 Environment Programming* book for more information on communications.

---

## UPDATE Procedure

The UPDATE procedure allows you to create or change a data file utility (DFU) program to change records in a disk file. File specifications define the format of the records in the file. For more information about DFU, see the *ADTS/400: Data File Utility* book.

```
UPDATE  file name,dfu program name,[file source member name],  
  
      [records],[[D],[Z],[B]],[[NN],[NY],[YN],[YY],[GO]], [dfu source member name],,  
  
      [library name],[display source member name],[name of file on disk]  
      [current library]
```

59020284-0

**file name** This specifies the file to be changed. The file name can be up to 8 characters.

### DFU program name

This specifies the DFU program used to process the file. If the program does not exist in the library, DFU starts the setup procedures to create the program. If the program exists in the library specified, DFU runs it. To change an existing DFU program, leave this name blank and specify the name of the program to be changed in the DFU source member name parameter. If both the DFU program name and the DFU source member name are not provided, DFU starts the setup procedures to create a temporary program.

See the "UPDATE# Procedure" on page 4-295 for alternate methods of running a DFU program.

### file source member name

This specifies the RPG II source member containing the file description (F-specifications) and record input specifications (I-specifications) that describe the file to be processed. This member can contain one or more sets of file description and input specifications, or an entire RPG II program. The file description and input specifications that correspond to the file are taken as the data description.

This parameter is prompted for if you do not specify it. It is required if the specified DFU program does not exist.

**records** This is supported for System/36 compatibility only. The value is not used.

**D, Z, or B**

This indicates whether unkeyed selected zoned numeric fields are to be filled with zeros (hex F0) or blanks. The only entries allowed are D, Z, or B. If no parameter is specified, D is assumed. D or B specifies a data file with blank fill of unkeyed selected zoned numeric fields. Z specifies a data file with zero fill of unkeyed selected zoned numeric fields.

**NN, NY, YN, YY, or GO**

This is supported for System/36 compatibility only. The value is not used.

**DFU source member name**

To change an existing DFU program, specify the name of the DFU program in this parameter and leave the DFU program parameter blank. If the DFU program is specified, then this parameter is ignored.

**library name**

This specifies the library that contains or will contain the DFU program. All library members associated with the DFU job are looked for or stored in this library. If you do not specify this parameter, the current library is assumed.

**display source member name**

This parameter is not supported. If a value is specified, an error message is issued.

**name of file on disk**

This specifies the name of the disk file to be changed (if it is different from the name specified in the DFU program). If you specify this parameter, you can have several programs that refer to different files change the same file disk. This is an optional parameter. If you specify the name of a file on disk and do not specify a file to be changed by the DFU program, you are prompted for that parameter.

**Example**

This example creates a program named FILEFMT to change a disk file named FILE1. The library MYLIB will contain the DFU program FILEFMT.

```
UPDATE FILE1,FILEFMT,,,,,MYLIB
```

---

## UPDATE# Procedure

The UPDATE# procedure allows you to change records in a disk file using the data file utility (DFU) by running the specified DFU program.

## UPDATE#

```
UPDATE# file name,dfu program name,[file source member name],  
  
[records],[D],[Z],[B],[NN],[NY],[YN],[YY],[GO],[dfu source member name],,  
0  
  
[library name],[display source member name],[name of file on disk]  
current library
```

59020674-0

**file name** This specifies the file to be changed. The file name can have up to eight characters.

### **DFU program name**

This specifies the DFU program to run to change records in the file. This program must already exist in the library.

### **file source member name**

This is supported for System/36 compatibility only. The value is not used.

**records** This is supported for System/36 compatibility only. The value is not used.

### **D, Z, or B**

This indicates whether unkeyed selected zoned numeric fields are to be filled with zeros (hex F0) or blanks. The only entries allowed are D, Z, or B. If no parameter is specified, D is assumed. D or B specifies a data file with blank fill of unkeyed selected zoned numeric fields. Z specifies a data file with zero fill of unkeyed selected zoned numeric fields.

### **NN, NY, YN, YY, or GO**

This is supported for System/36 compatibility only. The value is not used.

### **DFU source member name**

This is supported for System/36 compatibility only. The value is not used.

### **library name**

This specifies the library that contains or will contain the DFU specifications. All library members associated with the DFU job are looked for, or stored, in this library. If you do not specify this parameter, the current library is assumed.

### **display source member name**

This is supported for System/36 compatibility only. The value is not used.

### **name of file on disk**

This specifies the name of the disk file to be changed (if it is different from the name specified in the DFU program). If you specify this parameter, you can have several programs that refer to different files change the same file disk. This is an optional parameter. If you specify the name of a file on disk and do not specify a file to be changed by the DFU program, you are prompted for that parameter.

### Example

This example changes records in a disk file named FILE1. The library MYLIB contains the DFU program FILEFMT.

```
UPDATE# FILE1,FILEFMT,,,,,MYLIB
```

---

## WRKSPL Procedure

The WRKSPL procedure is supported only for compatibility with the IBM System/36. Only the procedure interface is supported. The displays and functions provided by the System/36 environment WRKSPL procedure are similar, but not identical, to those provided by the System/36 procedure.

The WRKSPL procedure runs the Work with Spooled Files (WRKSPLF) command, using the assistance level specified in your user profile. The WRKSPLF command provides the displays, and processes the options that you choose. See the WRKSPLF command in the *CL Reference* book for information on how to display and work with spool files.

The performance of this procedure depends on the number of spool file entries that exist on your system. The time that is required to display the list of spool files may be longer on systems with many spool file entries.

---

## WRKUSER Procedure

The WRKUSER procedure is supported only for compatibility with the IBM System/36. Only the procedure interface is supported. The displays and functions provided by the System/36 environment WRKUSER procedure are similar, but not identical, to those provided by the System/36 procedure.

The WRKUSER procedure runs the Work with User Jobs (WRKUSRJOB) command. The WRKUSRJOB command provides the displays, and processes the options that you choose. See the WRKUSRJOB command in the *CL Reference* book for information on how to display and work with user jobs.

---

## WSFLOAD Procedure

Not supported. On System/36, the WSFLOAD procedure installed the PC Support/36 work station feature support from diskette. See "Managing Licensed Programs" on page 4-2 for information on installing licensed programs.

---

## WSFSAVE Procedure

Not supported. On System/36, the WSFSAVE procedure saved the PC Support/36 work station feature support to diskette. See "Managing Licensed Programs" on page 4-2 for information on saving licensed programs.

---

### WSU Procedures

The following WSU procedures are not supported:

- WSULOAD
- WSUSAVE
- WSUTXCR
- WSUTXEX
- WSUTXRV

See the *System/36-to-AS/400 Work Station Utility Conversion Guide* book for information on converting your WSU programs.

---

### XREST Procedure

The IBM System/34 XREST procedure is not supported.

To do a similar function, use the Copy DBCS Font Table (CPYIGCTBL) command. The CPYIGCTBL command is only supported for the DBCS version of the OS/400 program.

See the *CL Reference* book for more information on the CPYIGCTBL command. See the *ADTS/400: Character Generator Utility* book for more information on DBCS font tables.

---

### XSAVE Procedure

The IBM System/34 XSAVE procedure is not supported.

To do a similar function, use the Copy DBCS Font Table (CPYIGCTBL) command. The CPYIGCTBL command is only supported for the DBCS version of the OS/400 program.

See the *CL Reference* book for more information on the CPYIGCTBL command. See the *ADTS/400: Character Generator Utility* book for more information on DBCS font tables.

---

## Chapter 5. OCL Statements

This chapter describes the System/36 operation control language (OCL) statements and how you can use them on the AS/400 system. The OCL statements allow you to run programs and control how the system runs programs. The following information appears for each statement:

- The function of the statement.
- The placement of the statement in relation to other statements in a procedure and the circumstances under which it is needed.
- The syntax format of the statement. For a description of the rules used to describe the syntax formats, see “Conventions Used for Describing Syntax Formats” on page 1-3.
- Descriptions of the statement's parameters.
- One or more examples of how to use the statement.

---

### Placement of OCL Statements

OCL statements make up a job or job step. The end of a job step is indicated by a RUN OCL statement or, if the program being run reads utility control statements, an END utility control statement. You must place certain OCL statements (for example, the COMM OCL statement) between the LOAD and RUN OCL statements for a job step. You can place other OCL statements anywhere among the OCL statements.

The following example shows OCL statements that make up two job steps. The first job step (which runs the ACTREC program) ends with the first RUN OCL statement. The second job step (which runs the \$COPY utility program) ends with the END utility control statement.

```
* First job step
// LOAD ACTREC
// FILE NAME-DATA
// RUN
*
* Second job step
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-DATA,DISP-SHR
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

---

### Types of Information Contained in OCL Statements

The OCL statements contain two types of information: an **identifier** and one or more **parameters**. The identifier distinguishes one OCL statement from another. A parameter supplies information to the System/36 environment. The general form of an OCL statement is:

```
// identifier parameter1,parameter2, . . .
```

## Identifiers

Every OCL statement except INCLUDE requires a statement identifier. All OCL statements begin with // followed by one or more blanks. For example, in the following statement:

```
// LOAD PROGRAM
```

the OCL statement identifier is LOAD. The program loaded is named PROGRAM.

The end of data statement:

```
/*
```

does not require the //.

## OCL Parameters

Parameters are either **positional** or **keyword** parameters. A positional parameter is a value. A keyword parameter is a keyword followed by a value. In the following statement, the **positional parameters** PROG1 and MYLIB are the name of a program to load and the library containing the program:

```
// LOAD PROG1,MYLIB
```

Enter positional parameters in the order shown in the syntax formats.

In the following statement, NAME-COPYIN and LABEL-FILE1 are **keyword parameters**:

```
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
```

A keyword parameter contains a **keyword** (NAME and LABEL are the keywords in the previous OCL statement) that distinguishes the parameter from other parameters, just as statement identifiers distinguish one OCL statement from another. In addition to a keyword, a keyword parameter usually contains a **value** (COPYIN and FILE1 are values in the previous OCL statement).

Enter keyword parameters in any order, not necessarily in the order shown in the syntax formats.

## Procedure Parameters

A procedure parameter can be any combination of single-byte character set (SBCS) characters. A procedure parameter cannot contain double-byte character set (DBCS) characters. Use question marks (?), commas (,), apostrophes ('), slashes (/), hyphens (-), greater than signs (>), equal signs (=), plus signs (+), and blanks with caution because they have special meanings within procedures.

If a parameter contains embedded blanks or commas (,), it must begin with an apostrophe and end with an apostrophe. A parameter enclosed in apostrophes is considered a character string. All data between the apostrophes processes as one parameter. The beginning and ending apostrophes are not processed as part of the parameter. The length of the parameter is set to equal the number of characters between the apostrophes. The following example shows the character string *June 17, 1985* as the first parameter for PROCA. The parameter length is 13.

```
PROCA 'June 17, 1985'
```

If you include an apostrophe within a character string, you must specify it as two apostrophes.

```
'one o'clock'
```

The length of the character string is set to equal the number of characters between the beginning and ending apostrophes, except that two apostrophes within the string are counted as one. The following example shows how you enter the character string *It's one o'clock*. The length of the character string is 16.

```
'It's one o'clock'
```

If a parameter does not begin with an apostrophe, any other apostrophes encountered in the parameter are considered data, not character strings. The following example shows parameters with apostrophes as data. The first parameter is *Mary's* and the second parameter is *o'clock*.

```
PROCA Mary's,o'clock
```

---

## General OCL Coding Rules

See “Conventions Used for Describing Syntax Formats” on page 1-3 for a description of how to enter procedures, commands, and OCL statements. The following are additional rules for OCL statements:

- Enter the // in positions 1 and 2.
- Enter one or more blank positions between the // and the statement identifier. For example:

```
// LOAD
```

or:

```
//  LOAD
```
- Enter one or more blank positions between the statement identifier and the first parameter. For example:

```
// LOAD PROG1,MYLIB
```

or:

```
//  LOAD  PROG1,MYLIB
```
- When including more than one parameter, use a comma to separate the parameters. No blanks are allowed within or between parameters. Anything following the first blank after a parameter is considered a comment (see “Comments” on page 5-4).
- When entering keyword parameters, place the keyword first and use a hyphen (-) to separate the keyword from the value.
- If you do not specify a value with a keyword parameter, the parameter is ignored. For example:

```
// FILE NAME-FILEA,DATE-,UNIT-F1
```

When this statement is processed, the DATE keyword parameter is ignored.

## Continuing OCL Statements

Expressing a single statement in two or more lines is called **continuation**. You can continue any OCL statement that contains keyword parameters.

Rules for using continuation are:

- Begin each new line with // in positions 1 and 2.
- Leave one or more blanks between the // and the first parameter in the line.
- Place a comma after the last parameter in every line except the last line. The comma, followed by a blank, tells the System/36 environment that the statement continues on the next line.

The following example of a continued FILE OCL statement uses four lines to express the statement:

```
// FILE NAME-INPUT,  
// LABEL-FILE1,  
// RECORDS-250,  
// RETAIN-J
```

The System/36 environment interprets the line as:

```
// FILE NAME-INPUT,LABEL-FILE1,RECORDS-250,RETAIN-J
```

See "Continuing the Lines of a Procedure" on page 2-7 for another way to continue OCL statements. This method uses a plus sign (+) in the last nonblank position in the line to indicate the line is continued. For instance, you could enter the preceding example as the following:

```
// FILE NAME-INPUT,+  
// LABEL-FILE1,+  
// RECORDS-250,+  
// RETAIN-J
```

It would be interpreted as:

```
// FILE NAME-INPUT,LABEL-FILE1,RECORDS-250,RETAIN-J
```

It takes the system longer to read continued statements.

If a record ends with a shift-in character just before the continuation expression, and the first nonblank character of the next record is a shift-out character, both the shift-in and shift-out characters are removed.

## Comments

Use comments to explain the purpose of the statements in a procedure. Comments in a procedure appear when you print a procedure on a printer or view it on a display station. Comments do not appear when the procedure runs. Comments can contain any combination of characters. However, the continuation symbol (+ sign) is recognized in a comment if the statement does not have an asterisk (\*) in column 1. You can include comments in the following places:

- Following the asterisk (\*) on the comment statement. For example:

```
* THIS IS AN EXAMPLE OF A COMMENT STATEMENT
```

The comment here is: THIS IS AN EXAMPLE OF A COMMENT STATEMENT.

- After the last parameter in a statement. Leave one or more blanks between the last parameter and your comment. For example:

```
// LOAD $COPY    LOAD THE COPY UTILITY
```

In this example, the comment is: LOAD THE COPY UTILITY.

Another example:

```
// RUN    RUN THE PROGRAM
```

The comment here is: RUN THE PROGRAM. The RUN statement has no parameters.

- After the comma that follows the last parameter in a continued line. For example:

```
// FILE NAME-INPUT,    FILE STATEMENT FOR
// LABEL-FILE1,    INPUT FILE
// RECORDS-250,
// RETAIN-J
```

In this example, the first two lines of the FILE statement contain comments.

- After the procedure name in an INCLUDE OCL statement if the statement has parameters but none of them are specified. Leave a blank after the procedure name, enter a comma, leave a blank after the comma, and enter the comment. For example:

```
PAYROLL , RUN PAYROLL PROCEDURE
```

or:

```
// PAYROLL , RUN PAYROLL PROCEDURE
```

or:

```
// INCLUDE PAYROLL , RUN PAYROLL PROCEDURE
```

The comment in these statements is: RUN PAYROLL PROCEDURE.

#### Notes:

1. Substitution expressions contained in a comment statement are evaluated, except comment statements preceded by an asterisk (\*).
2. An INCLUDE OCL statement that calls either a multiple requester terminal (MRT) procedure or a procedure that passes data, not parameters, cannot contain a comment.

---

## ABEND OCL Statement

This statement is supported for System/36 compatibility only. Syntax checking is done, but no other action is taken.

```
// ABEND    OUTPUT- { DISK
                   { DISKETTE
                   { PRINTER
                   { TAPE
```

59020292-0



For diskette, the AUTO parameter is supported for compatibility only. If specified for diskette, the value is syntax-checked, but is ignored when the diskette drive is allocated.

- YES** This specifies that if the end of tape is reached while processing a file, the next tape drive is automatically used. If the tape drive cannot be allocated, the original tape drive is used for the next tape.
- NO** This specifies no automatic advance to another tape drive when another tape needs processing.

**CONTINUE**

This is supported for compatibility only. If specified, the value is syntax-checked, but is ignored when processing the diskette.

CONTINUE does not apply for tape. If you specify CONTINUE for tape, an error message is issued.

**WAIT**

This specifies whether the procedure waits for the drive to become available. If you use the ALLOCATE statement, but omit the WAIT parameter, WAIT-YES is assumed.

- YES** This specifies that if the drive is already allocated or is in use, the procedure should wait for the drive to become available. The procedure waits for the period specified on the default wait (DFTWAIT) of the OS/400 class. After this, the procedure proceeds as if WAIT-NO was specified.  
  
See the *Work Management* book for more information on OS/400 class objects.
- NO** This specifies that if the drive is already allocated or is in use, the procedure should continue. Use the ?CD? substitution expression to see if the drive was allocated to the procedure. See “?CD? (Return Code) Expression” on page 3-13 for more information about the values substituted by this expression.

**Example 1**

The following examples create a series of diskette files containing libraries. The diskette has a volume ID of VOL001.

```
// ALLOCATE UNIT-I1
SAVELIBR MYLIB,,VOL001
SAVELIBR YOURLIB,,VOL001
SAVELIBR OURLIB,,VOL001
SAVELIBR PAYLIB,,VOL001
```

or:

```
// ALLOCATE UNIT-I1
// LOAD $MAINT
// FILE NAME-MYLIB,UNIT-I1,PACK-VOL001
// FILE NAME-YOURLIB,UNIT-I1,PACK-VOL001
// FILE NAME-OURLIB,UNIT-I1,PACK-VOL001
// FILE NAME-PAYLIB,UNIT-I1,PACK-VOL001
// RUN
// COPYLIBR FROM-MYLIB,TO-DISK,FILE-MYLIB
// COPYLIBR FROM-YOURLIB,TO-DISK,FILE-YOURLIB
// COPYLIBR FROM-OURLIB,TO-DISK,FILE-OURLIB
// COPYLIBR FROM-PAYLIB,TO-DISK,FILE-PAYLIB
// END
```

### Example 2

The following example allocates the diskette drive to a job and directs the procedure not to wait for the drive to become available. The diskettes to contain the files have a volume ID of VOL002.

```
// ALLOCATE UNIT-I1,WAIT-NO
// IFF ?CD?=2033 GOTO OK
// PAUSE 'Diskette drive allocated to another job.'
// CANCEL
// TAG OK
SAVE ALL,,,VOL002,ALL,S1
SAVELIBR LIBR1,,,VOL002,S1
SAVELIBR LIBR2,,,VOL002,S1
```

### Example 3

The following example allocates tape drive 1 to a job and directs the procedure not to wait for the drive to become available. The tape to contain the files is on tape drive 1 and has a volume ID of VOL002.

```
// ALLOCATE UNIT-T1,WAIT-NO,AUTO-YES
// IFF ?CD?=2036 GOTO OK
// PAUSE 'Tape drive 1 allocated to another job.'
// CANCEL
// TAG OK
SAVE ALL,,,VOL002,ALL,T1
SAVELIBR LIBR1,,,VOL002,,,T1
SAVELIBR LIBR2,,,VOL002,,,T1
```

---

## ATTR OCL Statement

The ATTR OCL statement does the following:

- Indicates whether an operator can end a job before the job has completed normally
- Indicates whether another job can start at a display station while a job is interrupted
- Changes the maximum number of requester terminals for a multiple requester terminal (MRT) program
- Changes the NEP indicator for a program
- Assigns a priority to a job or job step
- Releases the requester display station from the next job step when the next job step begins running
- Informs you if a job has ended normally or abnormally

**Placement:** Place the ATTR OCL statement anywhere among the OCL statements except between the LOAD and RUN OCL statements.

```
// ATTR [ CANCEL- {YES} ] [ INQUIRY- {YES} ] [ ,MRTMAX- nnn ] [ ,NEP- {YES} ]
        [ ,PRIORITY- {HIGH} ] [ ,RELEASE- {NO} ] [ ,NOTIFY- {NO} ] [ ,MRTWAIT- {YES} ]
          {YES}
          {NO}
          {NO}
          {NO}
          {YES}
          {NO}
          {NO}
          {NO}
          {YES}
          {NO}
          {NO}
          {NO}
          {YES}
          {NO}
```

59020294-3

**CANCEL** This specifies whether an operator can cancel a job by pressing the System Request key and choose either options 2 or 90 from the System Request menu. Choosing option 2 ends the job, and choosing option 90 signs the user off. This parameter only affects the options shown on the System Request menu. The operator can press the System Request key and temporarily interrupt the job, but options 2 or 90 to cancel the job are not available. The operator can still cancel the job by using the End Job (ENDJOB) command.

This condition remains in effect until another ATTR statement changes it, or until the job ends.

**YES** This specifies appearance of options 2 and 90 of the System Request menu. The operator cancels the job by taking either option 2 or 90.

**NO** This specifies that options 2 and 90 of the System Request menu do not appear. The operator cannot cancel the job using the System Request menu.

**INQUIRY** This specifies whether an operator can start another job by pressing the System Request key and taking option 1 from the System Request menu, or whether an operator can resume an alternate interrupted job by pressing the System Request key and taking option 1 from the menu. This parameter only affects the options displayed on the System Request menu. This means the operator can press the System Request key and temporarily interrupt the job, but option 1 is not available. This condition remains in effect until you change it with another ATTR statement, or until the job ends.

If you specify the INQUIRY parameter, the program's noninquirable indicator is ignored. The indicator is set by the RPG II compiler.

**YES** This specifies appearance of option 1 on the System Request menu. The operator starts another job by choosing option 1 and entering the command or menu number required to start another job.

**NO** This specifies that option 1 of the System Request menu does not appear. This prevents the operator from starting another job, or resuming an alternate interrupted job.

**MRTMAX** This specifies the number of requesters (display stations or ICF sessions) that can be attached to the multiple requester terminal (MRT) program run in the next job step. Only one ATTR statement in a job step can specify the MRTMAX parameter.

**nnn** This changes the MRTMAX value specified on the COMPILE statement when the program was compiled. nnn cannot exceed the MRTMAX value specified on the COMPILE statement or in the RPG II file description specification. Use MRTMAX only if you specified a MRTMAX value of one or more on the COMPILE statement. Do not enter leading zeros.

**NEP** This specifies whether the program is a NEP. NEP is a long-running program. Other jobs cannot use the system resources, except for shared files and the spool file, that are allocated to a NEP.

**YES** This specifies that the program is a NEP.

**NO** This specifies that the program is not a NEP.

For a MRT program, if you do not specify NEP-YES on the ATTR OCL statement and if NEP was not specified when the program was compiled, the program is not a NEP program. When a MRT program that is not a NEP releases its last attached device, the system checks the MRTDLY attribute of the MRT procedure to determine if the MRT should end immediately, or delay termination for a short period of time, in case a new requester would like to attach to the MRT. If the MRTDLY attribute indicates that the MRT should delay termination, the MRT will wait for the number of seconds specified for the System/36 environment before giving the program a return code instruction it to go to the end of the program. If the MRTDLY attribute indicates that the MRT should not delay termination, the program will immediately be given a return code instructing it to go to the end of the program. For more information, see the Procedures section of the *System/36 Environment Programming* book.

If you specify NEP-YES for a MRT program, the MRT program does not end when it releases its last attached display. Instead, the MRT program waits until another display requests it. The system operator can end the MRT program at any time by using the ENDJOB command.

## **PRIORITY**

This specifies the processing priority for a job or job step. The system assigns system resources in order of decreasing priority. The order of priority is HIGH or YES, MEDIUM or NO, and LOW or NO. For example, if you specify PRIORITY-MEDIUM, system resources are assigned to the job or job step after they are assigned to any other higher priority (HIGH or YES) job or job step, but before they are assigned to any lower priority (NO or LOW) job or job step. PRIORITY-YES and PRIORITY-HIGH are equivalent. PRIORITY-NO is equivalent to PRIORITY-MEDIUM for interactive, MRT, and ICF-evoked jobs, and equivalent to PRIORITY-LOW for batch jobs. You can specify the PRIORITY parameter more than once in a job. This parameter takes effect as soon as it is encountered. See the *System/36 Environment Programming* book for more information about job priority.

**RELEASE**

This specifies whether the display station remains allocated to the next job step. The RELEASE parameter is ignored for jobs on the job queue. If you do not specify RELEASE, RELEASE-NO is assumed.

**NO** This specifies that the next job step is not released and that the display station remains allocated to the job step.

**YES** This specifies the release of the display station when the next job step begins running. If that job step is the last or only step in a procedure, the command display appears at the display station and the operator can start another job. If that job step is not the last step in a procedure, only that job step releases. The released step runs at the same time as following steps of the same procedure. The requesting display station remains allocated to the steps that follow the released step. If you specify RELEASE-YES, consider the following:

- Do not pass existing job files to the released step. For information about job files, see “FILE OCL Statement (for Disk Files)” on page 5-25.
- Do not pass continued print files to the released step. For more information about continued print files, see “PRINTER OCL Statement” on page 5-68.
- Only those files referenced by FILE OCL statements between the following LOAD and RUN OCL statement pair are passed to the released job step. All other FILE OCL statements, including FILE OCL statements that have JOB-YES specified, continue to be owned by the original job and can be used in following job steps. This means that a job step released by an ATTR OCL statement cannot own files that are specified outside of that job step’s LOAD and RUN OCL statements.
- Scratch files pass to the released job step unless you specify a given file as a resident (by a FILE OCL statement with RETAIN-T specified) outside the LOAD and RUN OCL statement pair.
- Job files created by the released step are treated as scratch files. You cannot use those files by following steps in this procedure.
- The RELEASE-YES parameter is ignored if the OCL statements for the job step contain a WORKSTN statement that specifies REQD-YES for the requesting display station.
- System messages issued by the released step appear at the system console, not at the display station that released the job step.
- A released step uses a copy of the user program status indicator (UPSI) switches for the requesting display station and the display station local data area as they exist when the released step starts. If the released step

changes the display station local data area or the switches, the changes remain in effect only during the job step. You cannot see the changes by following steps in the procedure or by later jobs submitted from the display station.

- If you specify RELEASE-YES for a job step that runs a MRT program that is also defined as an NEP, the MRT program starts but has no display stations. The MRT program then waits for the next requesting display station.
- If you specify RELEASE-YES for a procedure evoked through the intersystem communications function (ICF), the ICF session is released from the evoked procedure.

**NOTIFY** This specifies whether you want a message when the job ends.

**JOB** This specifies that you receive a message when you start the job with:

```
// EVOKE,NRT(// ATTR RELEASE-YES),
or // JOBQ has ended.
```

**NO** This specifies that you receive no message.

#### **MRTWAIT**

This specifies if the operator should wait until the MRT is attached (if a request causes the number of MRTs to exceed the maximum). If you do not specify MRTWAIT, MRTWAIT-YES is assumed.

**YES** This specifies that the operator does not get back control until the MRT is attached.

**NO** This specifies that the operator gets back control, and receives a return code of 2045. Check this return code by using the ?CD? substitution expression.

#### **Example 1**

The following example prevents an operator from ending the payroll program by taking options 2 or 90 from the System Request menu:

```
// ATTR CANCEL-NO
// LOAD PAYROLL
// FILE NAME-EMPLOYES
// PRINTER NAME-PRINT,DEVICE-P2,FORMSNO-CHCK,ALIGN-YES
// RUN
```

#### **Example 2**

The following example assigns high processing priority to a job and releases it from the display station:

```
// ATTR PRIORITY-HIGH,RELEASE-YES
// LOAD PROG1
// RUN
```

## CANCEL OCL Statement

The CANCEL OCL statement cancels one or more spool file entries. Use the CANCEL OCL statement to cancel:

- All of your spool file entries
- All of your spool file entries with a specific forms number
- All of the spool file entries for your user ID

If you have \*JOBCTL authority, you can also cancel the following:

- All spool file entries for a specific printer
- All spool file entries for all printers
- All spool file entries with a specific forms number for all printers
- All spool file entries with a specific user ID for all printers

**Placement:** Place the CANCEL OCL statement anywhere among the OCL statements. You can evoke it or submit it to the job queue.

```
// CANCEL  PRT, {printer id
            (P) {ALL
                FORMS,forms number
                USER,user id
            }
}
```

59020615-0

**PRT or P** This specifies cancellation of one or more entries from the spool file.

**printer id** This specifies the 2-character ID of a printer for which all spool file entries are canceled. You must have \*JOBCTL authority to use this parameter.

**ALL** This specifies cancellation of all System/36 environment spool entries and all OS/400 spool entries to which you are authorized. You must have \*JOBCTL authority to use this parameter.

**FORMS** This specifies cancellation of the spool file entries with the specified forms number. If you have \*JOBCTL authority, all entries with the specified forms number are canceled for all printers you control. If you do not have \*JOBCTL authority, only your own spool file entries with the specified forms number are canceled.

**forms number**

This specifies the 1- to 4-character forms number of the spool file entries that are canceled.

**USER** This specifies cancellation of spool file entries with the specified user ID. If you specify your own user ID, all your spool file entries are canceled. If you specify a user ID other than your own, all spool file entries with the specified user ID are canceled for all printers. You must have \*JOBCTL authority to specify a user ID other than your own.

**user id** This specifies the 8-character user ID of the spool file entries that are to be canceled.

## CHANGE

### Example 1

The following example cancels all spool file entries printed on printer P2:

```
// CANCEL PRT,P2
```

or:

```
// CANCEL P,P2
```

### Example 2

The following example cancels all spool file entries with forms number 1234 for all printers if you have \*JOBCTL authority:

```
// CANCEL PRT,FORMS,1234
```

or:

```
// CANCEL P,FORMS,1234
```

---

## CHANGE OCL Statement

The CHANGE OCL statement changes entries on the spool file. Use the CHANGE OCL statement to change the following:

- The number of copies printed of your spool file entries
- The number of copies printed of all of your spool file entries with a specific forms number
- The forms number used for your spool file entries
- The new forms number used for all of your spool file entries with a specific forms number
- The printer used for your spool file entries
- The printer used for all of your spool file entries with a specific forms number

If you have \*JOBCTL authority, you can also change the following:

- The number of copies printed of all spool file entries with a specific forms number for all printers
- The number of copies printed of all spool file entries with a specific user ID for all printers
- The new forms number used for all spool file entries with a specific forms number for all printers
- The forms number used for all spool file entries with a specific user ID for all printers
- The printer used for all spool file entries from among the printers
- The printer used for all spool file entries with a specific forms number for all printers
- The printer used for all spool file entries with a specific user ID for all printers

**Placement:** Place the CHANGE OCL statement anywhere among the OCL statements. You can evoke it or submit it to the job queue.

```
// CHANGE { COPIES,copies,{FORMS,forms number }
           { USER,user id }
           FORMS,forms number,{FORMS,forms number }
           { USER,user id }
           ID,new printer id,{old printer id
                             FORMS,forms number }
                             { USER,user id }
```

59020616-0

- COPIES** This specifies a change in the number of copies printed for one or more spool file entries.
- copies** This specifies the number of copies printed and is any number from 1 to 255.
- FORMS** This specifies a change in the number of copies of spool file entries with the specified forms number. If you have \*JOBCTL authority, all entries with the specified forms number change for all printers you control. If you do not have \*JOBCTL authority, only your spool file entries with the specified forms number change.
- forms number**  
This specifies the 1- to 4-character forms number of the spool file entries for which the number of copies changes.
- USER** This specifies a change in the number of copies of spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries change. If you specify a user ID other than your own, all spool file entries with the specified user ID change for all printers. You must have \*JOBCTL authority to specify a user ID other than your own.
- user id** This specifies the 8-character user ID of the spool file entries for which the number of copies changes.
- FORMS** This specifies a change in the printout form for one or more spool file entries.
- forms number**  
This specifies the new 1- to 4-character forms number of the printout form used.
- FORMS** This specifies a change in the printout form for spool file entries with the specified forms number. If you have \*JOBCTL authority, all spool file entries with the specified forms number change for all printers. If you do not have \*JOBCTL authority, only your spool file entries with the specified forms number change.
- forms number**  
This specifies the current 1- to 4-character forms number of the spool file entries for which the printout form changes.
- USER** This specifies a change of the printout form for spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries change. If you specify a user ID other than your own, all spool file entries with the specified user ID change for all printers. You must have \*JOBCTL authority to specify a user ID other than your own.
- user id** This specifies the 8-character user ID of the spool file entries for which the printout form changes.
- ID** This specifies a change of the printer used for selected spool file entries.
- Note:** If you direct your printed output to another printer with different printer characteristics, this may result in printing or programming errors, or improper printing of your output. (Word processing jobs may not complete successfully.)

**new printer id**

This specifies the 2-character ID of the new printer used.

**old printer id**

This specifies the 2-character ID of the current printer. All spool file entries using this printer now use the new printer. If a spool file entry is currently printing on the old printer, it continues printing on that printer. You must have \*JOBCTL authority to use this parameter.

**FORMS** This specifies use of the new printer by spool file entries with the specified forms number. If you have \*JOBCTL authority, all spool file entries with the specified forms number change for all printers. If you do not have \*JOBCTL authority, only your spool file entries with the specified forms number change.

**forms number**

This specifies the 1- to 4-character forms number of the spool file entries for which the printer changes.

**USER** This specifies use of the new printer by spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries change. If you specify a user ID other than your own, all spool file entries with the specified user ID change for all printers that you control. You must have \*JOBCTL authority to specify a user ID other than your own.

**user id** This specifies the 8-character user ID of the spool file entries for which the printer changes.

**Example 1**

The following example changes the number of copies of printout for all spool file entries with user ID AA120199. Only one copy of these spool file entries is supposed to print, but you want three copies to print.

```
// CHANGE COPIES,3,USER,AA120199
```

**Example 2**

The following example changes the printout form for all spool file entries with current forms number 1324. These spool file entries were supposed to print on printout form 1324, but you want to print them on printout form 6978.

```
// CHANGE FORMS,6978,FORMS,1324
```

---

**COMM OCL Statement**

Not supported. The System/36 COMM OCL statement was used by batch BSC programs to assign a line number to a program using communications, and to assign a phone number list to a program using the autocall feature.

Use the BSCCEL SESSION OCL statement in place of the COMM OCL statement. See "SESSION OCL Statement" on page 5-84 for a description of the SESSION OCL. You will need to create the appropriate BSCCEL controller, line, and device description needed by ICF. See "LISTDONE (Phone List Completion) Condition" on page 3-34 for more information on phonelist. See the *System/36-Compatible*

*RPG II User's Guide and Reference* book for information on using telecommunications specifications in the System/36 environment.

## COMPILE OCL Statement

The COMPILE OCL statement supplies information required by the OS/400 program when a library source member is compiled. The COMPILE OCL statement does the following:

- Identifies the library source member that contains the source program compiled. A source program is a collection of statements, such as RPG II specifications, compiled into a running program.
- Identifies the library containing the source program.
- Identifies the library containing the compiled load member. Then you can load and run the load member using the LOAD and RUN OCL statements.
- Specifies the maximum number of requesting display stations attached to the program, if the program is a MRT program.
- Specifies whether the program is a NEP.
- Specifies the data dictionary used by the program.

**Placement:** Place the COMPILE OCL statement between the LOAD and RUN OCL statements of the job step that compiles the source program. If the source program is in the procedure or keyboard job stream, omit the COMPILE OCL statement.

```
// COMPILE SOURCE-name [ ,INLIB-{library name}
                        {current library} ]
                        [ ,OUTLIB-{library name}
                        {current library} ] [ ,MRTMAX-{nnn} ] [ ,NEP-{YES}
                        {NO} ]
                        [ ,DATADCT-data dictionary name ] [ ,MRO-{YES}
                        {NO} ]
```

59020296-1

**SOURCE** This specifies the name of the source member that contains the source program compiled.

**INLIB** This specifies the name of the library that contains the source program. If you specify INLIB, only that library is searched. If you do not specify INLIB, the current library is assumed.

**OUTLIB** This specifies the name of the library that will contain the compiled load member. If you do not specify OUTLIB, the current library is assumed. The name of the load member is either specified in the source program or is the same as the source program.

**MRTMAX** This identifies the program as an MRT program.

**nnn** This specifies the maximum number of requesting display stations attached to the program. Do not enter leading zeros.

If nnn is 0 or if you do not specify MRTMAX, the object program is not a MRT program. If you specify MRTMAX, change the maximum number of requesting terminals by using an ATTR statement when you run the object program.

## COMPILE

**NEP** This specifies whether the program is a NEP. NEP is defined as a long-running program.

**YES** This specifies that the program is a NEP.

**NO** This specifies that the program is not a NEP.

When the program runs, an ATTR statement can change the NEP indicator.

- For a MRT program, if you do not specify NEP-YES on the ATTR OCL statement, and if NEP was not specified when the program was compiled, the program is not a NEP program. When a MRT program that is not a NEP releases its last attached device, the system checks the MRTDLY attribute of the MRT procedure to determine if the MRT should end immediately, or delay termination for a short period of time, in case a new requester would like to attach to the MRT. If the MRTDLY attribute indicates that the MRT should delay termination, the MRT will wait for the number of seconds specified for the System/36 environment before giving the program a return code instructing it to go to the end of the program. If the MRTDLY attribute indicates that the MRT should not delay termination, the program will immediately be given a return code instructing it to go to the end of the program.
- If you specify NEP-YES for a MRT program, the MRT program does not end when it releases its last attached display. Instead, the MRT program waits until another display requests it. The system operator can end the MRT program at any time by using the End Job (ENDJOB) command.

For information about NEPs, SRTs, and MRTs, see the *System/36 Environment Programming* book.

### **DATADCT**

This specifies the name of the library containing the file definitions used by the program. On System/36, file definitions were stored in data dictionaries.

**MRO** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**Example**

The following example compiles the source member named PROG3 in the library named MYLIB. The compiled load member is placed in the system library. The program PROG3 runs as an NEP.

```
// COMPILE SOURCE-PROG3,INLIB-MYLIB,OUTLIB-#LIBRARY,NEP-YES
```

---

**DATE OCL Statement**

A DATE OCL statement entered anywhere among the OCL statements other than between the LOAD and RUN OCL statements changes the session date. (A session begins when an operator signs on, and normally ends when the operator enters the OFF command.) If no DATE statement or DATE procedure establishes a session date, the system date specified during initial program load is used as the session date.

A DATE OCL statement entered between LOAD and RUN OCL statements specifies the program date (also known as the job step date). When the job step ends, the date is set back to the session date. If you do not enter a DATE statement, or do not run the DATE procedure between a LOAD statement and a RUN statement, the session date is used as the program date. If there are two or more DATE statements between a LOAD statement and a RUN statement, the last DATE statement is used.

**Notes:**

1. The program date is used to determine how long a diskette file is retained, and is printed on output. The program date is also used as the creation date of any disk files created by the program.

The session and program dates are also affected by the CHGJOB CL command. For more information on handling dates in a System/36 environment job, see the job date and date format section in the *System/36 Environment Programming* book.

2. If a job is placed on the job queue, the program date when the job was placed on the queue is assigned to the job.
3. At 2400 hours, the system date updates automatically, but the session date and program date are not updated for active sessions or programs.
4. The DATE OCL statement affects the AS/400 job date. The job date is changed to the new session date when the DATE OCL statement is outside a LOAD/RUN pair. When the DATE OCL statement is between the LOAD and RUN OCL statements, the job date is changed to the new program date. After the RUN OCL statement is processed, the job date is changed back to the session date.
5. The System/36 date format is set from the AS/400 date format if possible. If the AS/400 date format is Julian, the System/36 date format cannot be set from it, since the System/36 environment does not support the Julian date format. In this case, the AS/400 date format is changed to yymmdd, as is the System/36 session and program date. A message is logged that the AS/400 date format was changed.

**Placement:** Place a DATE statement anywhere among the OCL statements.

## DATE

```
// DATE { mddy }  
          { dmyy }  
          { yymm }
```

59020297-0

The date specified on the DATE statement must be in the current session date format. Use any of the following three formats for the session date:

- Month-day-year (mddy)
- Day-month-year (dmyy)
- Year-month-day (yymm)

Use the STATUS command to determine the current session date format, and the SET procedure to change the current session date format.

Month, day, and year must each be 2-digit numbers, but you can omit leading zeros in month and day when you use punctuation. In the punctuated form, use any characters except commas (,), apostrophes ('), numbers, question marks (?), and blanks as punctuation.

Enter the date with or without punctuation. For example, specify July 24, 1983 in any of the following ways:

Date	Date Format
7-24-83	mm-dd-yy
24-7-83	dd-mm-yy
83/7/24	yy/mm/dd
072483	mddy
240783	dmyy
830724	yymm

A date of all zeros (000000) is invalid.

### Example 1

The following examples specify the DATE statement for July 1, 1983:

```
// DATE 070183
```

or:

```
// DATE 7-1-83
```

### Example 2

The following example specifies a program date (job step date) for the PAYROLL program by placing the DATE statement between the LOAD and RUN statements for the PAYROLL procedure. When the PAYROLL program ends, the session date is used for the PAYPRNT program.

The PAYROLL program reads the EMPLOYES file and calculates the paychecks. The checks include the date. By specifying the program date using the DATE statement, you can run the program before the actual payroll date (indicated by parameter 1, the ?1R? expression). The PAYPRNT program does not use the date.

The example PAYROLL procedure contains the following statements:

```

* PAYROLL procedure
* Calculate paychecks using date specified
// LOAD PAYROLL
// DATE ?1R?
// FILE NAME-EMPLOYES,DISP-OLD
// FILE NAME-CHECKS,RETAIN-J,BLOCKS-30
// RUN
*
* Print check information on check forms (CHCK)
* Allow the checks to be aligned in the printer
// LOAD PAYPRNT
// FILE NAME-CHECKS,RETAIN-J
// PRINTER NAME-PRINT,SPOOL-NO,LINES-25,FORMSNO-CHCK,
//          ALIGN-YES,DEVICE-P3
// RUN

```

To run the procedure and specify a payroll date of 7-1-84, enter:

```
PAYROLL 7-1-84
```

and the DATE statement processes as:

```
// DATE 7-1-84
```

---

## DEALLOC OCL Statement

The DEALLOC OCL statement frees the diskette drive or tape drives after an ALLOCATE OCL statement has allocated the drive. This allows other jobs on the system to use the drive.

**Placement:** Place the DEALLOC statement anywhere among the OCL statements.

```

// DEALLOC UNIT- { I1
                  T1
                  T2
                  'T1,T2'
                  TC
                }

```

59020298-1

<b>UNIT</b>	This specifies which drive is freed for other programs to use.
<b>I1</b>	This specifies the freeing of the diskette drive.
<b>T1</b>	This specifies the freeing of tape drive 1.
<b>T2</b>	This specifies the freeing of tape drive 2.
<b>'T1,T2'</b>	This specifies the freeing of both tape drive 1 and tape drive 2. Enclose this option within single quotes and separate them by a comma.
<b>TC</b>	This specifies the freeing of the tape cartridge.

### Example

This example allocates the diskette drive for three procedures and then frees it before program PROG1 is loaded so that other jobs can use the drive:

```
// ALLOCATE UNIT-I1,WAIT-YES
SAVE FILE1,,,VOL001
SAVE FILE2,,,VOL001
SAVE FILE3,,,VOL001
// DEALLOC UNIT-I1
// LOAD PROG1
// RUN
```

---

## DEBUG OCL Statement

The DEBUG OCL statement specifies whether the procedure control expressions and OCL statements in your procedures should list as they are evaluated, and whether the procedures should stop after each job step. This allows you to follow the logic path of your procedures.

**Placement:** Place the DEBUG OCL statement anywhere among the OCL statements. If you enter the DEBUG OCL statement from the keyboard, the specified parameters remain in effect until you enter another DEBUG OCL statement, or until you sign off the system by entering the OFF command.

If you specify the DEBUG OCL statement in a *procedure*, it remains in effect until another DEBUG OCL statement processes, or until the procedure ends. When the procedure ends, the parameters entered at the *keyboard* (if any) take effect.

```
// DEBUG [ PROC- $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} ] [ ,STEPHALT- $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} ]$$ 
```

59020299-0

**PROC** This specifies whether the procedure control expressions and the OCL statement should be listed as they are evaluated. The information is listed on the system list device and in the job log. To determine your present system list device, enter the STATUS SESSION command. The SYSLIST procedure sets the system list device.

If the system list device is the display station, and the listing of the job step shown does not fill the display, no message displays indicating the end of system list data before the next job step is shown. Other system list output generated by your procedure could combine with the procedure control expressions and OCL statements.

**NO** This specifies that listing does not occur. If PROC is not specified, NO is assumed.

**YES** This specifies that listing occurs.

**Note:** The output goes to the system list device in effect when the DEBUG OCL statement was encountered. That system list device remains in effect until you turn debugging off. Any SYSLIST procedure or OCL statement, or any PRINTER OCL statement having the NAME-\$SYSLIST parameter, is ignored if debugging is on.

### STEPHALT

This specifies whether the procedure is halted (that is, temporarily stopped) after each job step in the procedure.

- NO** This specifies that the procedure runs normally. If you do not specify STEPHALT, NO is assumed.
- YES** This specifies that a message displays after each job step in the procedure completes. The message indicates the name of the first-level procedure (if you are running a procedure) and the name of the program run, and the operator can select an option either to continue or to cancel the procedure.

### Example

The following example uses the DEBUG statement to view the procedure control expressions used during a sample procedure named SAMPLE:

```
* SAMPLE Procedure
* Parameter 1: File name (required, prompted for if omitted)
* Parameter 2: File's creation date (optional)
// * 'SAMPLE PROCEDURE RUNNING'
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-?1R?,
//   IFF ??= DATE-???,
//   UNIT-F1
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

The following statements run the procedure and produce the debug listing:

```
// DEBUG PROC-YES
SAMPLE FILE1
```

The DEBUG statement produces the following output:

```
SAMPLE FILE1
* SAMPLE Procedure
* Parameter 1: File name (required, prompted for if omitted)
* Parameter 2: File's creation date (optional)
// * 'SAMPLE PROCEDURE RUNNING'
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-?1R?,
//   FILE NAME-COPYIN,LABEL-FILE1,
//   IFF ??= DATE-???,
//     IFF = DATE-???,
//   UNIT-F1
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

---

## EVOKE OCL Statement

Use the EVOKE OCL statement to evoke (start) a procedure. When a procedure is evoked, it begins running as a separate job, and control returns immediately to the calling procedure. Thus, several procedures can run at once as a result of several EVOKE OCL statements. The EVOKE OCL statement is useful for jobs that require no operator input or response, or for jobs whose output is not required by a subsequent job step.

When a job is evoked, the priorities of the evoked job and the job that evoked it are the same.

**Note:** Procedures that are evoked and not started before the next IPL do not complete normally.

Messages issued from the evoked job are sent to the QSYSOPR message queue. When a procedure is evoked, the evoked procedure uses copies of the UPSI switches and the local data area as they exist on the requesting display station. If the evoked procedure changes the UPSI switches or the local data area, the changes are in effect only for the evoked job. Later jobs run or evoked at the requesting display station are not affected by the changes. When a procedure is evoked, the current files library is used as the initial files library for the evoked procedure. If the current files library is changed by the evoked procedure, it does not affect the original job.

The EVOKE OCL statement cannot evoke a MRT procedure.

Unexpected errors can occur if an evoked job contains any of the following:

- Setting the system list device to the CRT (SYSLIST CRT), then running one of the following procedures or equivalent utility control statements:
  - CATALOG
  - LISTFILE
  - LISTDATA
- Running the LISTDATA procedure with CRT specified for the output parameter

**Note:** Using too many EVOKE OCL statements might slow down the system.

**Placement:** Place the EVOKE statement anywhere among the OCL statements except between the LOAD and RUN OCL statements.

```
// EVOKE  procedure name [ ,library name ] [ parm1,parm2...
                                     *ALL ]
```

#### **procedure name**

This specifies the procedure evoked.

#### **library name**

This specifies the library in which to search for the procedure. If the specified library does not contain the procedure, the system library (#LIBRARY) is searched. If you do not specify a library name, the current library is searched first, and then the system library (#LIBRARY).

#### **parm1,parm2, ...**

This specifies parameters for the procedure. Parameters are not allowed if you specified PDATA=YES when the procedure member was created. The parameters may or may not be required, depending on the procedure they are passed to.

A parameter is any combination of characters except commas or blanks. Use question marks (?), apostrophes ('), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) with caution because they have special meanings within procedures. The total number of characters for one parameter must not exceed 128. An EVOKE statement passes a maximum of 64 parameters, separated by commas.

When entering parameters, type up to 512 characters. For example, type thirty-two 16-character parameters or sixty-four 8-character parameters. However, the combined total length of *all* parameters cannot exceed 1024 characters. You can accomplish this length by using substitution expressions and the local data area. See “Continuing the Lines of a Procedure” on page 2-7 for information about how to continue input lines to get more than 120 characters.

See “INCLUDE OCL Statement” on page 5-49 for more information about how to enter parameters. Do not specify program data.

**\*ALL** This specifies the passing of all parameters from the current procedure level to the procedure being called. Specify *\*ALL* only within a procedure. If you specify *\*ALL* as the only parameter and it is entered from the keyboard or selected by a menu item, an error message displays. If you specify *\*ALL* as one of the parameters, it is treated as a single parameter.

### Example

In the following example, the OCL statements include an EVOKE OCL statement that calls and passes parameters to the procedure PROC1. When PROC1 is evoked, it starts running and control immediately returns to the calling procedure. Thus, both procedures run at the same time and independently.

```
// EVOKE PROC1 parm1,parm2
// LOAD PAYROLL
// RUN
```

**Note:** Procedures that are evoked and not started before the next IPL will not complete normally.

---

## FILE OCL Statement (for Disk Files)

The FILE OCL statement for disk files supplies the System/36 environment with information about disk files. The system uses this information to read records from and write records to the disk file.

For information about the OS/400 program disk file concepts, see the *System/36 Environment Programming* book.

**Placement:** Each disk file created or used by a program requires a FILE OCL statement. Place the FILE OCL statement anywhere among the OCL statements.

If you place the FILE statement before a LOAD statement, the system immediately attempts to make the user the owner of the disk file. (See the JOB keyword for information on the duration of file acquisition.) Acquiring a disk file in this way establishes only the ownership level of the file (that is, which programs can use the file), either shared or old. The file is not actually created or used until it is allocated, opened, and used by a program (some programming languages combine the allocate and open steps into one operation). This allows you to determine, before starting a series of job steps, whether all the files required by the jobs steps are available.

If you place the FILE OCL statement between the LOAD and the RUN OCL statements for a job step, the file is acquired when the RUN OCL statement processes.

## FILE (Disk)

```
// FILE NAME-file name [ ,UNIT-F1 ] [ ,LABEL-file label ] [ ,RECORDS-records ]
[ ,BLOCKS-blocks ]

[ ,LOCATION- { A1
              A2
              A3
              A4
              block number } ] [ ,RETAIN- { T
                                           J
                                           S } ] [ ,DATE- { mmddyy
                                                         ddmmyy
                                                         yymmdd } ]

[ ,DISP- { SHR
           SHRMM
           SHRMR
           SHRRL
           SHRRR
           NEW
           OLD } ] [ ,JOB- { YES
                             NO } ] [ ,WAIT- { YES
                                                NO } ] [ ,EXTEND-value ]

[ ,DFILE- { NO
           YES } ] [ ,BYPASS- { NO
                               YES
                               PRF } ] [ ,DUPKEYS- { NO
                                                       YES } ]

[ ,DBLOCK-records ] [ ,IBLOCK-index entries ]

[ ,STORINDX- { YES
              NO
              maximum storage index size } ]
```

RV2P176-0

**NAME** This specifies the name the program uses to refer to the file. Use any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. Use question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) with caution because they have special meanings within procedures. The first character of a file name must be alphabetic (A through Z, #, \$, or @). Do not exceed 8 characters.

**UNIT-F1** This specifies that the file is on the disk. Do not specify this parameter for a disk file because F1 is the assumed value for the UNIT parameter.

**LABEL** This specifies the actual name or label by which a file is identified on the disk. Only specify the LABEL parameter when the actual name of the disk file is different from the name used in the program. If you omit the LABEL parameter from a disk FILE statement, the file name from the NAME parameter is used. If a program does not refer to an existing file by the file's actual name on the disk, you must supply a LABEL parameter.

File names are any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. Use question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) with caution because they have special meanings within procedures. The first character of a name must be alphabetic (A through Z, #, \$, or @). Do not exceed 8 characters.

The name could be a file that belongs to a file group. If the file name contains a period, the file is a member of a file group. The characters in front of the period identify the file group. See "SAVE Procedure" on page 4-220 for information about file groups.

### RECORDS

This specifies the number of records to reserve for the file. The total reserved space is rounded up to the next block (one block contains 2560 bytes), allowing enough space to contain at least the number of

records indicated. The smallest disk file unit that can be reserved is one block. For example, if you specify ten 50-byte records (500 bytes of space are needed for the file), then 2560 bytes (one block) are reserved.

RECORDS cannot exceed the value 8000000.

Put either RECORDS or BLOCKS, but not both, in the FILE OCL statement.

Use the RECORDS or BLOCKS parameter for a new file.

**BLOCKS** This specifies the number of disk blocks to reserve for the file. There are 2560 bytes in one disk block.

BLOCKS cannot exceed 512815.

Put either RECORDS or BLOCKS, but not both, in the FILE OCL statement.

Use the RECORDS or BLOCKS parameter for a new file. Use the CATALOG procedure to determine the number of disk blocks available for files.

#### LOCATION

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**RETAIN** This specifies the file classification as a resident (T), job (J), or scratch (S) file when the file is created or referred to.

If the RETAIN parameter is omitted from the FILE statement when the file is created or referred to, the file is assumed resident. However, if there is an existing job file with the same label, the job file is used.

**T** This specifies a resident file. A resident file remains on the disk when the job ends. The area containing a resident file becomes available for another file only under one of the following conditions:

- You supply a FILE OCL statement containing RETAIN-S for the resident file, to change the file retention to scratch. If the program uses the file, the file does not exist after the program ends normally. If the program does not end normally, the files do not change to scratch files and still exist as resident files. If the file is not used by the program, the file remains a resident file after the program ends.
- Another file with the same label loads into the area occupied by the resident file, changing the data or the organization of the file. You must specify the DISP-OLD parameter.
- You use the DELETE procedure to delete the file.

The actual number of files that can be placed on the disk depends on the size of the disk and the size of the files.

**J** This specifies a job file. A job file, after it is created, can be used by all of the remaining job steps in a job. The job file is defined only within the job and does not exist after the job

ends. If you release a job step (using the RELEASE parameter on the ATTR OCL statement) or that job step runs a MRT program, that step cannot use job files defined by other steps of the job. Any job files created by a released job step or a job step that runs a MRT program are treated as scratch files and cannot be used by other steps in the job. Two or more jobs using job files with the same name can run at the same time because the job files are defined only within the individual jobs.

When you create a file with RETAIN-J specified, following steps in the job can refer to that file by not specifying a RETAIN parameter or by specifying RETAIN-J or RETAIN-S. If a following step specifies RETAIN-J or does not specify RETAIN, the file is kept for later use by other job steps. However, if a following step specifies RETAIN-S for an existing job file, that file is removed at the end of that step and cannot be used by following steps in the job.

If a file is created with RETAIN-J, a following job step can use or create another disk file with the same label by specifying RETAIN-T. You cannot process a resident disk file in the same job step with a job file having the same label.

**Note:** If a system failure occurs, you lose the contents of a job file or a new scratch file.

**S** This specifies a scratch file. The scratch file does not exist after the job step ends.

**DATE** This specifies the creation date of an existing file. Use this parameter to ensure that the proper version of a file is used. The DATE parameter is not allowed when you specify DISP-NEW.

When a file is created on disk, its label and creation date are written on the disk as identification. The job step date when the file is created is the date used as the creation date. If date-differentiated files were specified during system configuration, you can give more than one resident file the same label. However, the creation dates of these files must be different. To refer to such a file, specify its label and date. If the date is not specified, the file having the latest date is used. The System/36 environment system collects date-differentiated files as members in a file. These members must all have the same file characteristics (record length, DFILE, EXTEND, DUPKEY values must be the same).

Enter the date in one of the three following formats:

- Month-day-year (mmdyy)
- Day-month-year (ddmmyy)
- Year-month-day (yymmdd)

However, the format you choose must be the same as the session date format. Use the STATUS SESSION command to determine the session date format.

### **DISP (Disposition)**

This specifies the file as new, old, or able to be shared by other jobs running on the system. You cannot use the DISP parameter if you

specify RETAIN-J. If you do not specify DISP, the system determines whether a file is new or old based on whether the file is already on disk. This is not the same, however, as assigning NEW or OLD as the DISP parameter. Specify DISP-OLD to overlay a file, even if the system determines that the file is old.

- SHR** This specifies that the file already exists and that other programs running on the system can share the file. You can perform read, update, delete, and add operations on the file. SHR is the same as SHRMM.
- For a description of programming considerations for file sharing, see the *System/36 Environment Programming* book.
- SHRMM** This specifies that the program using the file can modify the file (read, update, delete, or add records). Other programs sharing the file can also modify the file. SHRMM is the same as SHR.
- SHRMR** This specifies that only the program using the file can modify the file (read, update, delete, or add records). Other programs that are sharing the file can only read records from the file.
- SHRRM** This specifies that the program using the file only needs to perform read operations on the file (no records are updated, deleted, or added). Other programs that are sharing the file can modify the file (read, update, delete, or add records to the file).
- SHRRR** This specifies that the program using the file only needs to perform read operations on the file (no records are updated, deleted, or added). Other programs sharing the file can also only read records from the file.
- NEW** This specifies the file as new. It reserves the requested file name within the System/36 environment. If a file already exists with the same label and creation date as the new file, an error message appears. The new file is created using any disk file organization. No other programs can share the file until the program that created it ends.
- OLD** This specifies that the file already exists, and is not to be shared until the program using it ends. If the file does not exist, an error message appears. DISP-OLD allows you to process an existing file as an output file.
- If the old file attributes do not match the file attributes specified on the FILE statement, then the system deletes the old file and creates a new file with the new attributes. The link to a data dictionary is removed when the old file is deleted.
- If you are using a date-differentiated file, then the file attributes on the FILE statement must match the old file attributes. If the attributes do not match, an error message appears.

**JOB** This specifies whether the disk file is acquired for the entire job. If you specify JOB-NO, or if a file is not already acquired and you have not specified a JOB parameter, the file remains acquired only until the job step ends.

**YES** This specifies file acquirement for the entire job. The share level specified by the DISP parameter is used throughout the job.

You can only specify JOB-YES on a FILE OCL statement that is outside of a LOAD and RUN OCL statement pair.

If you specify JOB-YES, the other specified parameters are assumed each time the programs in the following job steps use the same file. For example, if you use the same file in a following job step, you need not specify another FILE OCL statement for that job step.

If you encounter a subsequent FILE OCL statement with the same NAME as a previous FILE OCL statement with JOB-YES specified, the subsequent FILE OCL statement must have the same LABEL as the JOB-YES FILE OCL statement, or must not specify a LABEL.

If a subsequent FILE OCL statement with the JOB-YES parameter is encountered that has the same NAME parameter as a previous FILE OCL statement having JOB-YES specified, and the file has not yet been created, all parameters specified on the current FILE OCL statement permanently override the original defaults. Any parameters that you do not specify on the current FILE OCL statement but that you did specify on the previous FILE OCL statement are lost. This, in effect, replaces the original parameter defaults.

If a subsequent FILE OCL statement without the JOB-YES parameter is encountered with the same NAME parameter as a previous JOB-YES FILE OCL statement, and the file has not yet been created, all specified parameters override the original default specifications. Any parameters you do not specify on the current FILE OCL statement default to what you specified on the previous JOB-YES FILE OCL statement. When the current job step ends, all the original default specifications for the file (except LOCATION, BLOCKS, and RECORDS) again become the parameter defaults if the file was not created. The LOCATION, BLOCKS, and RECORDS parameters used to create the file are always from the most recently processed FILE OCL statement.

You can specify JOB-YES for a new file. The specified file is acquired for exclusive use of the job and cannot be shared. The file is not created by specifying JOB-YES. The file is created when a program allocates, opens, and uses it. Some programming languages combine the allocate and open into one operation. If you specify the LOCATION, BLOCKS, or RECORDS parameters, they are only used when the file is created.

If you specify RETAIN-S and JOB-YES on a FILE OCL statement, only the job step creating the scratch file can use the value you specify, and the file does not exist after the job step ends.

If you specify an ATTR OCL statement with the RELEASE-YES parameter, only those files referred to by FILE OCL statements between the following LOAD and RUN OCL statement pass to the released job step. All other FILE OCL statements, including FILE OCL statements that have JOB-YES specified, continue to be owned by the original job and can be used in following job steps. This means that a job step released by an ATTR OCL statement cannot own files that are specified outside of that job step's LOAD and RUN OCL statements.

When you specify JOB-YES for a single requester terminal (SRT) procedure, and that procedure calls a MRT procedure that uses the same file, the share levels must be compatible on both FILE OCL statements in both procedures.

- NO** This specifies use of that file for only that job step.
- WAIT** This specifies whether the procedure should wait for the file to become available.
- YES** This specifies that if another job has already acquired the file, the procedure should wait for the file to become available.
- NO** This specifies that if another job has already acquired the file, the procedure should continue. Only specify NO on a FILE OCL statement that is outside of a LOAD and RUN OCL statement pair. Use the ?CD? substitution expression to see if the file was acquired by the procedure. See “?CD? (Return Code) Expression” on page 3-13 for more information about the values substituted by this expression.
- EXTEND** This specifies a file extension value to use when creating a new file. If this is not a new file, this keyword is ignored. If an extend value is not specified, an extend value is calculated. This calculated value is the number of records the file is extended by each time additional space is needed. The algorithm used to calculate this value is  $32\ 768$  divided by the record length of the file. For example, if the record length of a file is 100, the file is extended by 327 records ( $32\ 768/100 = 327$ ).
- value** This specifies a 1- to 8-digit block or record value indicating the amount of additional space to use for the extension. When the file is created, the size of the file is specified in either blocks or records. Use the same unit of measure here. The extension should be large enough to contain at least one record. If you specify EXTEND-0, no file extension occurs.
- On add operations for sequential and indexed files, the file is extended when the current size of the file is not large enough.
- Note:** When files are shared, if one user causes an extension to occur, all users of that file can take advantage

of the additional file space, whether or not they specified EXTEND on their FILE OCL statements.

**DFILE** This specifies whether the file is delete-capable. With a delete-capable file, use the delete operation in your program to delete a record from the file. The DFILE parameter is allowed only for new files or existing files that are to be completely reloaded with new information.

**NO** This specifies that the file is not delete-capable. The programs using this file can perform no delete operations.

**YES** This specifies that the file is delete-capable.

If you specify DFILE-NO for an existing delete-capable file, or if you specify DFILE-YES for an existing file that is not delete-capable, a message may appear, and you can either continue or cancel the job.

**BYPASS** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**Note:** If you want to prevent shared open from occurring automatically, you can code the BYPASS-PRF. For more information about shared opens, see the *System/36 Environment Programming*.

### DUPKEYS

This specifies whether duplicate keys are allowed for a new indexed file. Once a file is created, you cannot override this keyword. If you specify the DUPKEYS parameter for a file that is not a new indexed file, the DUPKEYS parameter is ignored.

**NO** This specifies that the file cannot contain duplicate keys. If an attempt is made to add a record that has the same key as a record in the file, an error occurs. If you do not specify a parameter, DUPKEYS-NO is assumed.

**YES** This specifies that the file can contain duplicate keys. Checking for duplicate keys in the index is not performed even if you specify BYPASS-NO.

**DBLOCK** This is used to specify a blocking value to substitute for the system default blocking value. This value is only used for files accessed sequentially. See the *System/36 Environment Programming* book for more information on blocking records.

Records must be a number from 1 to 65 535. However, if a value greater than 32 767 is specified, a blocking value of 32 767 is used.

**IBLOCK** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

### STORINDX

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

### Example 1

A program creates a disk file with the following facts assumed:

- The name the program uses to refer to the file is OUTPUT.
- The label of the file on the disk is TRANSACT.
- The file contains 200 records and is extended by 100 records whenever the file becomes full.

- The system chooses the disk area to contain the file.

The following FILE statement defines the file:

```
// FILE NAME-OUTPUT, LABEL-TRANSACTION, RECORDS-200, EXTEND-100
```

### Example 2

A program writes over (overlays) the contents of an existing disk file with the following facts assumed:

- The disk file is labeled IVENTORY.
- The file is located at block number 500 and is 12 blocks long.
- The file name used by the program is OUTFILE.
- The file is opened (in the program) for output.

The following FILE statement defines the file:

```
// FILE NAME-OUTFILE, LABEL-IVENTORY, RETAIN-T, DISP-OLD
```

### Example 3

A program is to create a delete-capable file named FILE1.

The following FILE statement defines the file:

```
// FILE NAME-FILE1, DFILE-YES, RECORDS-250
```

### Example 4

The following example allows both programs, PROG1 and PROG2, to use the file TRANSACTIONS:

```
// FILE NAME-TRANSACTION, JOB-YES, DISP-SHRMM, WAIT-NO
// IF ?CD?=0000 GOTO OK
// * 'The file is being used, do you want to wait (Y or N)?'
// IF ?64R?=N RETURN
// FILE NAME-TRANSACTION, JOB-YES, DISP-SHRMM, WAIT-YES
// TAG OK
// LOAD PROG1
// RUN
// LOAD PROG2
// RUN
```

### Example 5

The following example uses the DBLOCK parameter to specify that a larger number of records are transferred than was specified in the program:

```
// LOAD PAYROLL
// FILE NAME-PAYROLL, DISP-OLD, DBLOCK-30
// RUN
```

---

## FILE OCL Statement (for Diskette Files)

The FILE OCL statement for diskette files supplies the System/36 environment with information about a diskette file. The System/36 environment uses this information to read from and write to the diskette. For more information about System/36 diskette file concepts, see the *System/36 Environment Programming* book.

The diskette cannot contain two files with the same label.

## FILE (Diskette)

The following table shows the System/36 environment utility programs that use FILE OCL statements for diskette files:

Utility Program	Program Description	Procedures
\$BICR	Basic data exchange	TRANSFER
\$COPY	Diskette file copy and display	LISTDATA, LISTFILE, ORGANIZE, RESTORE, SAVE
\$DUPRD	Diskette copy	COPY11
\$MAINT	Library maintenance	BLDLIBR, FROMLIBR, JOBSTR, RESTLIBR, SAVELIBR, TOLIBR

**Placement:** Each diskette file created or used by a program requires a FILE OCL statement. Place the FILE OCL statement for diskette files after the LOAD OCL statement and before the RUN OCL statement.

```
// FILE NAME-file name,UNIT-I1 [ ,LABEL-file label ]  
  
[ ,RETAIN- $\left\{ \begin{array}{l} \text{retention days} \\ \underline{1} \end{array} \right\} ] [ ,DATE- $\left\{ \begin{array}{l} \text{mmdyy} \\ \text{ddmmyy} \\ \text{yyymmdd} \end{array} \right\} ] [ ,PACK-volume id ]  
  
[ ,LOCATION- $\left\{ \begin{array}{l} \text{S1} \\ \text{S2} \\ \text{S3} \\ \text{M1.nn} \\ \text{M2.nn} \end{array} \right\} ] [ ,AUTO- $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} ]$$$$ 
```

59020302-0

**NAME** This specifies the name that the program uses to refer to the file. The file name is any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. However, use question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) with caution because they have special meanings within procedures. The first character of a file name must be alphabetic (A through Z, #, \$, or @). Do not exceed 8 characters in a file name.

**UNIT-I1** This specifies that the file is on one or more diskettes. Specify UNIT-I1 for a diskette file. If you omit the UNIT parameter, UNIT-F1 is assumed.

**LABEL** This specifies the actual name or label by which the file is identified on the diskette. If you omit the LABEL parameter from a diskette FILE statement, the name specified by the NAME parameter is used. If the file is an existing file, a LABEL parameter is required when the name the program used to refer to the file differs from the name that identifies the file on the diskette.

The name is any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. However, use question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) with caution because they have special meanings within procedures. The first character of a name must be alphabetic (A through Z, #, \$, or @). Do not exceed 8 characters in a name.

Date-differentiated files are not supported as individual files on a diskette. However, when a SAVE ALL operation is performed using the SAVE procedure or the \$COPY utility, all disk files which have the same label are saved in a single file on the diskette. In all cases, the system displays a warning message before it creates a file with the same name as an existing file on the diskette. You can then do one of the following:

- Insert another diskette and then allow the system to create the file on that diskette.
- Cancel the job.

For more information on the SAVE procedure, see "SAVE Procedure" on page 4-220.

**RETAIN** This specifies the number of days to retain a file. The RETAIN parameter is used to calculate an expiration date. Whenever you specify RETAIN for a file, the system determines the expiration date of the file by adding the system date to the number of days specified by the RETAIN parameter.

#### **retention days**

This can be any number from 0 through 999. If you do not specify RETAIN when a new file is created, one day is assumed. If you specify any number up to 998, the file is retained for the specified number of days. If you specify 999, the file is considered permanent and you can delete it only with the DELETE procedure.

When the system creates a diskette file, the system writes the creation date and the calculated expiration date of the file in international format (yymmdd). If an existing nonpermanent diskette file is referred to by a FILE OCL statement with a RETAIN parameter, the expiration date of the file is changed to the date determined by the RETAIN parameter.

If a FILE OCL statement with a nonpermanent RETAIN parameter refers to an existing permanent diskette file, a message appears and you can either cancel the job or continue processing. If you decide to continue processing after the message appears, the file remains a permanent file.

Whenever the system creates a file on a diskette or adds to an existing file on a diskette, all other files on the diskette whose expiration dates are the same as or earlier than the job step date and all expired files are deleted automatically. If a file being added to has expired, it is not deleted.

**DATE** This specifies the creation date of an existing file and ensures that the proper file is processed. When using the \$COPY utility to restore a file from diskette, this specifies the date that the disk file, saved within the diskette file of the same label, was originally created on disk.

If you specify both a name and a creation date on the FILE OCL statement, the system displays a message if it cannot find a file with that name and date on the inserted diskette. You can then insert another diskette, and the system looks for the specified file on that diskette.

When a file is created on diskette, its name, expiration date, and creation date (program date) are written on the diskette as identification. Both the creation date and the expiration date are written in the international format (yymmdd). The system converts the program date to the international format before writing it on the diskette. To ensure correct

## FILE (Diskette)

processing, files created by other systems should be created so that the creation date and the expiration date are written in the international format.

**PACK** This specifies the volume ID of the diskette, and is required when a program creates a file or adds to a file on a diskette. The INIT procedure puts the volume ID on the diskette. The volume ID is any combination of 6 or less alphanumeric characters.

The volume ID specified by the PACK parameter is compared with the volume ID of the diskette. If they are not the same, a message appears. You can then continue processing (the volume ID is ignored), insert the correct diskette, or cancel the job.

If you do not supply the PACK parameter on the diskette FILE statement for a diskette file that is being created or added to, a message appears and you must cancel the job.

The PACK parameter is not required for a diskette file being read. However, you should ensure that the proper diskette is inserted.

### LOCATION

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

Valid values for the diskette slot location are:

- S1, S2, or S3
- M1 or M2
- M1.nn or M2.nn (nn is a magazine slot number, from 1 to 10)

**AUTO** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

Valid values for diskette automatic advance are:

- YES
- NO

### Example

The following example shows how the \$COPY utility program uses FILE OCL statements to copy a disk file to diskette. Assume the following conditions:

- The name of the disk file is TRANS1, which is to be the name of the diskette file.
- The \$COPY program refers to the input file (the disk file in this example) as COPYIN. It refers to the output file (the diskette file) as COPYO.
- The diskette has a volume ID of VOL001.
- The diskette file is retained for 8 days.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-TRANS1
// FILE NAME-COPYO,UNIT-I1,LABEL-TRANS1,RETAIN-8,PACK-VOL001
// RUN
```

## FILE OCL Statement (for Tape Files)

The FILE OCL statement for tape files supplies the system with information about a tape file. The system uses this information to read from and write to the tape. For more information about tape file concepts, see the *System/36 Environment Programming* book.

The following table shows the System/36 environment utility programs that use FILE OCL statements for tape files:

Utility Program	Program Description	Procedures
\$TCOPY	Basic data exchange	TAPECOPY
\$COPY	Tape file copy and display	LISTDATA, LISTFILE, RESTORE, SAVE
\$MAINT	Library maintenance	FROMLIBR, RESTLIBR, SAVELIBR, TOLIBR, BLDLIBR, JOBSTR

**Placement:** Each tape file that a program creates or uses requires a FILE OCL statement. Place the FILE OCL statement for tape files after the LOAD OCL statement and before the RUN OCL statement.

```
// FILE      NAME-file name [ ,SEQNUM-number ] [ ,LABEL-file label ]

           [ ,REEL- $\left\{ \begin{array}{l} SL \\ NS \\ NL \\ BLP \end{array} \right\}$  ] [ ,VOLID- $\left\{ \begin{array}{l} \text{volume id} \\ \text{'volume id, volume id, ... volume id'} \end{array} \right\}$  ]

           [ ,UNIT- $\left\{ \begin{array}{l} T1 \\ T2 \\ TC \end{array} \right\}$  ] [ ,RETAIN- $\left\{ \begin{array}{l} \text{retention days} \\ \underline{1} \end{array} \right\}$  ] [ ,DATE- $\left\{ \begin{array}{l} \text{mddy} \\ \text{ddmmyy} \\ \text{yyymmdd} \end{array} \right\}$  ]

           [ ,RECFM- $\left\{ \begin{array}{l} F \\ V \\ FB \end{array} \right\}$  ] [ ,BLKL-block length ] [ ,RECL-record length ]

           [ ,DENSITY-1600 ] [ ,AUTO- $\left\{ \begin{array}{l} YES \\ NO \end{array} \right\}$  ] [ ,END- $\left\{ \begin{array}{l} \text{REWIND} \\ \text{LEAVE} \\ \text{UNLOAD} \end{array} \right\}$  ]
```

59020303-1

**Note:** The AUTO parameter is ignored for systems with only one tape drive.

**NAME** This specifies the name the program uses to refer to the file. Use any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. However, use question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) with caution because they have special meanings within procedures. The first character of a file name must be alphabetic (A through Z, #, \$, or @). Do not exceed 8 characters.

**SEQNUM** This specifies the placement of the file when a tape contains more than one file. SEQNUM is any number from 1 to 9999. For example, if the input file you want to use is the fifth file on the tape, specify SEQNUM-5 to access the file. Use this parameter, as opposed to the LABEL

parameter, with nonstandard labeled and nonlabeled tapes, and with tapes with bypass label processing.

If you specify SEQNUM and you also specify the LABEL parameter for an input file, the FILE OCL statement processes the SEQNUM parameter first. The header label name from the file found by the SEQNUM parameter is compared to the label name specified in the FILE statement. If the label names are the same, processing continues. If they are different, the FILE statement then looks for the file using the LABEL parameter.

If you specify neither the SEQNUM parameter nor the LABEL parameter, the file name is used as the label name.

**LABEL** This specifies the actual name or label by which the file is identified on the tape. If you omit the LABEL parameter from a tape FILE statement, the name specified by the NAME parameter is used. If the file is an existing file, you must use a LABEL parameter when the name the program used to refer to the file differs from the name that identifies the file on the tape. The LABEL parameter is valid only with standard label tapes. If the REEL parameter is not REEL-SL, the LABEL parameter is ignored.

Use any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), asterisks (\*), and blanks. However, use question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) with caution because they have special meanings within procedures. The first character of a name must be alphabetic (A through Z, #, \$, or @). Do not exceed 17 characters.

When two or more files on a tape have the same name, the system displays a warning message before it creates a file with the same name as an existing file on the tape. You can then do one of the following:

- Allow the system to create the file.
- Insert another tape and then allow the system to create the file on that tape.
- Cancel the job.

If two or more tape files have the same label specified on a FILE statement and you do not specify the DATE parameter, the program processes the *first file* on the tape with the specified label. If two or more tape files have the same label and creation date, the program processes the first file on tape with the specified label and creation date. To process a subsequent tape file with the same label and creation date, specify the file placement with the SEQNUM parameter. Unlike during disk processing, there is no search for the file with the most recent date.

**REEL** This specifies the type of label processing performed for the tape.

**SL** This specifies standard tape labels. If you do not specify REEL, SL is assumed.

**NS** This specifies nonstandard tape labels. If you specify UNIT-TC, REEL-NS is not valid.

**NL** This specifies nonlabeled tape. If you specify UNIT-TC, REEL-NL is not valid.

**BLP** This specifies the bypassing of the label processing, but the tape must have standard labels. If you specify UNIT-TC, REEL-BLP is not valid.

**Note:** If NS, NL, or BLP is specified for an input file, the operation will be prompted for the number of reels to process.

**VOLID** This specifies the volume ID of the tape or tapes in process. The VOLID parameter is valid only for tapes with standard labels (REEL-SL). The name contains up to 6 alphanumeric characters (A to Z, 0 to 9, @, #, \$).

**volume id**

This specifies the volume ID of the tape in process. If a multivolume output file is in process and you specify only one volume ID in the VOLID parameter, the volume ID of the first tape processed must match the volume ID you specify. The volume IDs of the second and subsequent tapes are not checked and do not have to match the specified volume ID. You must specify at least one volume ID for a multivolume output file.

If a multivolume input file is in process and you specify only one volume ID in the VOLID parameter, the volume ID of the first tape processed must match the volume ID you specify. The volume ID of the second and subsequent tapes does not have to match the volume ID of the first tape.

**'volume id,volume id,...volume id'**

This specifies that if a multivolume file is in process and you specify more than one volume ID in the VOLID parameter, the volume IDs must be enclosed within single quotes (') and separated by a comma (.). For example, 'vol1,vol2,vol3'. Specify up to 30 volume IDs. The volume ID of each tape processed must then match and be in the same order as the volume IDs you specify.

**UNIT** This specifies the tape drive on which the processing begins. Specify UNIT-T1, UNIT-T2, or UNIT-TC for a tape file. If you omit the UNIT parameter, UNIT-F1 is assumed.

**T1** This specifies that tape processing begins on the T1 device.

**T2** This specifies that tape processing begins on the T2 device.

**TC** This specifies that tape processing begins on the TC device.

**RETAIN** This specifies the number of days a file is retained. Use the RETAIN parameter to calculate an expiration date for standard label tapes only. Once written to tape, the date cannot be changed. Whenever you specify RETAIN for a file, the system determines the expiration date of the file by adding the program date to the number of days specified by the RETAIN parameter.

**retention days**

This can be any number from 0 through 999. If you do not specify RETAIN when a new file is created, one day is assumed. If you specify any number up to 998, the file is retained for the specified number of days. If you specify 999, the file is considered permanent.

## FILE (Tape)

When the system creates a tape file, the system writes the creation date and the calculated expiration date of the file in Julian format (cyydd). If you refer to an existing tape file with a FILE OCL statement that has a RETAIN parameter, the RETAIN parameter is ignored.

The expiration date is not checked when the tape file is allocated, but only when a tape is initialized (by the TAPEINIT procedure) and when writing between tape volumes. Only the first file on the tape is checked against the expiration date.

When a new file is created on a tape, the new file is placed after the last file on the tape.

**DATE** This specifies the creation date of an existing file and ensures that the proper file is processed. When using the \$COPY utility to restore a file from tape, this specifies the date that the disk file, saved within the tape file of the same label, was originally created on disk.

If you specify both a name and a creation date on the FILE OCL statement, the system displays a message if it cannot find a file with that name and date on the inserted tape. You can then mount another tape, and the system looks for the specified file on that tape.

When a file is created on tape, its name, expiration date, and creation date (program date) are written on the tape as identification. When a tape file is created, the system writes both the creation date and the expiration date in the Julian format (cyydd). The system converts the program date to the Julian format before writing it on the tape. If you specify the DATE parameter for an existing tape file, the system converts the Julian date to the session date format before using the file. To ensure correct processing, files created by other systems should be created so that the creation date and the expiration date are written in the Julian format.

**RECFM** This specifies the format of the input and output file records. If you do not specify the RECFM parameter, F is assumed unless you are reading or adding to a standard label tape file (REEL-SL) and the file in process contains a HDR2 label.

**F** This specifies the format to be fixed-length, unblocked records. If you specify UNIT-TC, RECFM-F is not valid.

**V** This specifies the format as variable-length, unblocked records (valid only for input files). If you specify UNIT-TC, RECFM-V is not valid.

**FB** This specifies the format as fixed-length, blocked records.

### **BLKL-block length**

This specifies the number of bytes in a physical block of data on tape. The parameter is any value from 18 to 32 767 bytes. Only use the BLKL parameter for fixed-length blocked records (specify the RECFM-FB parameter). The following files require the BLKL parameter:

- All output files with fixed-length blocked records.
- All input files with fixed-length blocked records not being processed by standard labels with an HDR2 label. (REEL-NS, REEL-NL, REEL-BLP, or REEL-SL and the file being processed do not have an HDR2 label.)

**RECL-record length**

This specifies the number of bytes in a logical data record on tape. The parameter is any value from 18 bytes to 4096 bytes. The RECL parameter is required by input files not processed by standard labels with an HDR2 label. That is, REEL-NS, REEL-NL, REEL-BLP, or REEL-SL and the file in process do not have an HDR2 label specified. If you specify the RECL parameter for a standard label tape (you specify REEL-SL and the file in process has an HDR2 label), it must match the record length contained within the tape label. If it does not match, an error message is issued.

**DENSITY-1600**

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**AUTO**

This specifies how tape files contained on more than one tape are processed. Use the ALLOCATE OCL statement to override this parameter. See "ALLOCATE OCL Statement" on page 5-6 for more information.

**YES**

This specifies that if you run out of tape while processing a file, an attempt is made to continue processing the file by allocating the other tape drive. Both tape drives must be available when tape processing begins. Otherwise, only the first tape drive is used.

AUTO-YES is ignored if you specify TC for the UNIT keyword. If you do not specify the AUTO parameter, AUTO-YES is assumed.

**NO**

This specifies that an attempt should not be made to continue a file on the next tape drive. If you run out of tape while processing a file, you will be prompted to mount the next tape on the specified drive.

**END**

This specifies the position of the tape after the file processes.

**REWIND**

This specifies the position to which the tape rewinds after processing completes.

If you specify REWIND for a reel-to-reel tape, the tape rewinds to the load point.

If you specify REWIND for a tape cartridge, the tape rewinds to the beginning of the cartridge.

**LEAVE**

This specifies that the tape remains in the position of the last use.

If you specify LEAVE for a reel-to-reel tape or a tape cartridge, the tape remains in the position of the last use. The next operation to the tape begins at that point.

When a job step using tape successfully completes and you specified LEAVE, the next job step using tape can then take advantage of the position of the tape in the following ways:

- If the next job step writes a new standard labeled tape file, the system checks to ensure that the tape is positioned at the end of the tape before writing the data to tape.

## FILE (Tape)

- If you specify a sequence number, the system searches from the beginning of the tape for the specified sequence number, validating that the sequence number can be written on this tape. If a tape file exists at that sequence number, it is written over by the new tape file.
- If the next job step reads a standard labeled tape file, the system validates that it is positioned to the correct file by checking the file label and the creation date (if specified). If the file label and creation date do not match, the system rewinds and searches the tape for the specified file. If you specify a sequence number, the system searches for the specified sequence number, performing validation of the file label and the creation date before reading the data.
- If the next job step reads a standard labeled tape file in bypass label processing mode, the system searches for the specified sequence number.
- If the next job step reads or writes to a nonlabeled tape file and you do not specify a sequence number, the system begins reading or writing data with no checking. If you specify a sequence number, the system rewinds the tape, then locates the sequence number.
- If the next job step reads a nonstandard labeled tape file, the system rewinds and reads the first file's data.

The procedure control expressions (PCE) of VOLID and DATAT cause the tape to rewind and search, starting from the beginning of the tape.

**UNLOAD** This specifies that the tape is rewound and unloaded after the file is processed.

If you specify UNLOAD for a reel-to-reel tape, the tape is rewound and unloaded after processing is completed.

If you specify UNLOAD for a tape cartridge, the tape is positioned to the *end* of the cartridge after processing is completed.

Each time a new or different tape cartridge is processed, or the latch on the tape drive is opened and closed, the cartridge must be prepared for operation. The tape is wound to the end, then rewound to the beginning. If REWIND or LEAVE was specified for the last operation, winding and rewinding the tape cartridge is a lengthy process. You can shorten the process by specifying UNLOAD, because the tape remains at the end after the last operation is completed.

### Example

This example shows how the \$COPY utility program uses the FILE OCL statements to copy a disk file to tape on tape drive 1. The name of the disk file is TRANS1, which is the name of the tape file. The \$COPY program refers to the input file (the disk file in this example) as COPYIN. It refers to the output file (the tape file) as COPYO. The tape has a volume ID of VOL001. If the file runs out of tape, the other tape drive does not finish copying the file. After the file is copied, the tape is rewound and unloaded for removal. The tape file is retained for 8 days.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-TRANS1
// FILE NAME-COPYO,LABEL-TRANS1,REEL-SL,VOLID-VOL001,+
//   UNIT-T1,RETAIN-8,+
//   AUTO-NO,END-UNLOAD
// RUN
```

---

## FILELIB OCL Statement

Use the FILELIB OCL statement to set the values for:

- The current files library
- The session files library
- The current library list search indicator
- The session library list search indicator

The following list describes how the files library initially is set:

**The current files library** initially is set from the session files library.

**The session files library** initially is set from the default files library in the System/36 environment configuration.

**The default files library** initially is set using the Change System/36 Environment Configuration (CHGS36) command.

The following list describes how the search indicators initially are set:

**The current library list search indicator** initially is set from the session library list search indicator.

**The session library list search indicator** initially is set from the default library list search indicator in the System/36 environment configuration.

**The default library list search indicator** initially is set using the Change System/36 Environment Configuration (CHGS36) command.

All files created by the System/36 environment functions are created in the current files library. All System/36 environment functions search the current files library for database files. However, if the current library list search indicator is set to search the library list, the system searches the library list for the specified files.

The current files library and the current library list search indicator remain in effect until you do the following:

- Sign off the system.
- Leave the System/36 environment by specifying the ENDS36 command.
- Use the FILELIB OCL statement with SESSION-NO to change the files library or the library list search indicator.
- Reach the end of a procedure in which the FILELIB OCL statement with SESSION-NO had been specified to change the files library or library list search indicator.
- Reach the end of a job in which either the FILELIB OCL statement with SESSION-YES or the FLIB procedure had been specified to change the files library or library list search indicator.

- Enter from the keyboard a FILELIB OCL statement with SESSION=YES to change the files library or the library list search indicator.
- Enter from the keyboard a FLIB procedure to change the files library or the library list search indicator.

**Placement:** Place the FILELIB OCL statement anywhere among the OCL statements. Multiple FILELIB OCL statements may be used in a job step. However, the current files library or the current library list search indicator (in effect at the time the FILE OCL statements are processed) determine which libraries are searched to locate the database files. Issuing another FILELIB OCL statement does not have an effect on previously processed FILE OCL statements.

```
// FILELIB [NAME- library name] [ ,LIBL- NO  
YES ] [ ,SESSION- NO  
YES ]
```

SP020681-1

**NAME** This specifies the name of the library that is to become the files library. If you specify NAME-0, the default files library from the System/36 environment configuration is used to change the files library. If the NAME keyword is not specified, the files library remains unchanged.

**LIBL** This specifies the value for the library list search indicator. If the parameter is not specified, the value remains unchanged from its previous value.

**NO** This specifies that the library list search indicator should be set so the library list is not searched for database files. Only the current files library is searched.

**YES** This specifies that the library list search indicator should be set so the library list is searched for database files. The current files library is not searched unless the files library is in the library list.

**SESSION** This specifies whether the values specified for the NAME and LIBL keywords are to be used to change the current or session values.

**NO** This specifies that the values specified for the files library (NAME) or the library list search indicator (LIBL) change the current values. The changes take effect immediately, and they remain in effect until the current procedure ends or until the current values are changed again. If you do not specify a SESSION parameter, NO is assumed.

**YES** This specifies that the values specified for the files library (NAME) or the library list search indicator (LIBL) change the session values. The changes take effect after the current System/36 job ends (the outermost procedure ends). They remain in effect until the session ends or the session values are changed again.

**Note:** If the FILELIB OCL statement is entered from the keyboard, the value specified for the SESSION parameter is ignored, and both the current and session values are changed.

**Note:** At least one of the two parameters, NAME or LIBL, must be specified. If both parameters are specified, both values are used and have the same effect as if two separate statements are used.

### Example 1

The following example indicates that a search is to be performed on the library list for database files:

```
// FILELIB LIBL=YES
```

### Example 2

The following example specifies that the library MYLIB is the current files library and that a search is to be performed on the library list for database files:

```
//FILELIB NAME-MYLIB,LIBL=YES
```

### Example 3

The FILELIB OCL statements in the following example specify:

- The PAYLIB1 library as the current files library. The library list is not searched for the file named PAYFILE1.
- The PAYLIB2 library as the current files library. The library list is not searched for the file named PAYFILE2.

```
// FILELIB NAME-PAYLIB1,LIBL-NO
// FILE NAME-PAYFILE1
// LOAD PAYROLL
// FILELIB NAME-PAYLIB2,LIBL-NO
// FILE NAME-PAYFILE2
// RUN
```

Since the files library is placed in the library list when the System/36 environment is started (always following library QSSP), changing the files library also changes the library list. When the // FILELIB OCL statement is issued, the user portion of the library list changes as follows:

- When the previous files library is already in the user library list, the new files library replaces it. Some exceptions are:
  - If the previous files library is QSSP or #LIBRARY, the new files library is inserted in the library list after library QSSP. If QSSP is in the system portion of the library list, the new library is added to the beginning of the user library list.
  - If the user adds the previous files library to the library list (for example, by using ADDLIBLE), the previous files library is left in the library list and the new files library is inserted in the library list following library QSSP.

**Note:** To remove the previous files library from the library list, it must have been placed in the library list by System/36 environment system functions either when entering the System/36 environment (initial library list is created) or by processing a // FILELIB statement. When leaving the System/36 environment, any files library added by the System/36 environment system function remaining in the library list is removed.

- If the previous files library is not in the user library list, the new files library is inserted in the library list after library QSSP.

- If the new files library is already in the library list (system or user), the previous files library is removed from the library list using the rules described previously, but no new library is added to the library list.
- When a procedure ends, the session files library is placed back in the library list if it differs from the current files library. This follows the rules described previously.

---

## FORMS OCL Statement

The FORMS OCL statement specifies information about the printer used during a display station session. Use the FORMS OCL statement to set the printing format for jobs that contain several printing steps without having to specify the same information several times on PRINTER OCL statements. You can specify the following information with the FORMS OCL statement:

- The printer ID of the printer used for all printed output
- The number of lines printed per page
- The forms number
- The horizontal print density (characters per inch)
- The vertical print density (lines per inch)
- The orientation or size of the printed output on the page (rotation or reduction)
- The printer drawer from which paper is used

Changes made by the FORMS OCL statement remain in effect until the display station session ends, or they are changed by one of the following:

- Another FORMS statement
- The PRINTER statement (for that job step only)
- The LINES procedure
- The PRINT procedure

A job placed on the job queue uses the values in effect when the job is placed on the queue. However, if a procedure running from the job queue contains a FORMS OCL statement, the job uses the FORMS OCL statement in the procedure.

**Placement:** You can place the FORMS OCL statement anywhere among the OCL statements.

```
// FORMS [DEVICE- $\left\{ \begin{array}{l} \text{printer id} \\ \text{SYSTEM} \end{array} \right\}$ ] [ ,LINES-value ] [ ,FORMSNO-forms number ]

[ ,CPI- $\left\{ \begin{array}{l} 10 \\ 15 \end{array} \right\}$ ] [ ,LPI- $\left\{ \begin{array}{l} 4 \\ 6 \\ 8 \end{array} \right\}$ ] [ ,ROTATE- $\left\{ \begin{array}{l} 0 \\ 90 \\ 180 \\ 270 \\ \text{COR} \end{array} \right\}$ ] [ ,DRAWER- $\left\{ \begin{array}{l} 1 \\ 2 \\ 3 \end{array} \right\}$ ]
```

59020304-1

If you use the FORMS statement, specify at least one parameter.

**DEVICE** This specifies the printer used for printouts. All printed output from the session prints on the specified printer.

If you do not specify a DEVICE parameter, the printer assignments do not change. The DEVICE parameter of the PRINTER OCL statement can override this parameter.

**printer id** This specifies the work station ID of the printer used. Use the STATUS WORKSTN command to determine the printer IDs.

**SYSTEM** This specifies use of the system printer.

**LINES** This specifies the number of print lines per page. The maximum number of lines per page is 112. If you do not specify the LINES parameter, and the number of lines per page was not previously set during the session, the system uses the value specified for the display station during system configuration, by the SET procedure, or by the \$SETCF utility program. If a program uses a line counter specification, that specification remains in effect only for the duration of that program.

For most user-written programs and System/36 environment utility programs that use the system list device for output, the following is true:

- Printing starts on line 6 of the first page and on line 6 of all new pages.
- If the value specified for LINES is greater than 12, printing skips to a new page when six lines less than the number of lines specified is printed.

For example, if you specify LINES-66, printing starts on line 6 and skips to a new page after printing line 60. Therefore, you get 55 lines of output per page ( $66 - 11 = 55$ , 5 blank lines at the top of a page and 6 blank lines at the bottom). If you specify LINES-13, two lines are printed per page. When you specify 12 or fewer lines per page, printing occurs on every line of the page.

For System/36 environment utility programs that use OS/400 print files for output, the following is true:

- Printing starts on line 1 of the first page and on line 1 of all new pages.
- If the value specified for LINES is greater than 12, printing skips to a new page when six lines less than the number of lines specified is printed.

For example, if you specify LINES-66, printing starts on line 1 and skips to a new page after printing line 60. Therefore, you get 60 lines of output per page ( $66 - 6 = 60$ , 6 blank lines at the bottom of the page). If you specify LINES-13, seven lines are printed per page. When you specify 12 or fewer lines per page, printing occurs on every line of the page.

**Note:** For a list of the System/36 environment utilities that use the system list device and the OS/400 print files, see the *System/36 Environment Programming* book.

For print operations from your programs, the System/36 environment indicates an overflow condition when six less than the number of lines specified (either in the program or in the FORMS statement) are printed.

#### FORMSNO

This specifies the forms number of the printer forms used for printed output from the display station session. (Each type of form should have a unique, user-assigned form number.) The form number is any combi-

nation of up to 4 characters except commas (,), apostrophes ('), or blanks.

**Note:** Use question marks (?), slashes (/), equal signs (=), plus signs (+), and hyphens (-) with caution because they have special meanings within procedures.

If you specify a forms number, the OS/400 program prompts the operator controlling the printer to install the forms with the specified forms number in the printer if the specified forms are not already installed.

**CPI** This specifies the horizontal print density to use for printed output from the display station session. Specify 10 or 15. If you use 15 characters per inch on another printer, a message appears and the operator controlling the printer either cancels or continues the printing.

Use the PRINTER statement to change the CPI value for a particular job step. This value stays in effect only for that job step. If you do not specify the CPI parameter and the CPI value was not previously set during the session, the system uses the value that was set when the printer was configured (either 10 or 15). (Because output can be directed to any printer from the spool file, this default configuration value is determined when the output is printed, not when it is intercepted in the spool file.) If no CPI value was specified when the printer was configured, a switch on the printer controls the CPI value.

**LPI** This specifies the vertical print density to use for printed output from the display station session.

Use the PRINTER statement to change the LPI value for a particular job step. This value stays in effect only for that job step. If you do not specify the LPI parameter, and the LPI value was not previously set during the session, the system uses the value set when the printer was configured. Because output can be directed to any printer from the spool file, this default configuration value is determined when the output is printed, not when it is intercepted in the spool file. LPI value must be specified when the printer is configured.

**ROTATE** This specifies rotation or size reduction of printed output on the page. It is valid only if the printer supports rotation of printed output.

- 0 This specifies that printed output rotates 0 degrees on the page.
- 90 This specifies that printed output rotates 90 degrees on the page.
- 180 This specifies that printed output rotates 180 degrees on the page.
- 270 This specifies that printed output rotates 270 degrees on the page.
- COR This specifies size reduction of the printed output. Output printed on 14-inch-wide paper prints on 8-1/2 x 11-inch paper.

**DRAWER** This specifies the drawer from which a printer selects paper.

- 1 This specifies that output is printed on paper from drawer 1. If the DRAWER parameter is not specified, 1 is assumed.

- 2 This specifies that output is printed on paper from drawer 2.
- 3 This specifies that output is printed on paper from drawer 3.

### Example 1

The FORMS statement in the following procedure specifies forms length for the job as 20 lines per page. It also specifies use of form CHEC.

```
// FORMS LINES=20,FORMSNO-CHEC
// LOAD PRNTCHEC
// PRINTER NAME-CHECKS,SPOOL=NO,ALIGN=YES
// RUN
```

### Example 2

The following FORMS statement specifies:

- A work station ID of P3 for the printer.
- A vertical print density of 8 lines per inch.
- A forms length of 11 inches (27.9 cm). Therefore, the lines per page is 88 (11 x 8 = 88).
- A horizontal print density of 15 characters per inch.

The names of the programs run are TEST and TEST1.

```
// FORMS DEVICE=P3,LINES=88,LPI=8,CPI=15
// LOAD TEST
// RUN
// LOAD TEST1
// RUN
```

---

## IMAGE OCL Statement

This statement is supported for System/36 compatibility only. Syntax checking is done, but no other action is taken.

```
// IMAGE { {MEMBER},print belt member name }
          { MEM }
          { XLATE,translation table member name }
```

S4620305-0

---

## INCLUDE OCL Statement

The INCLUDE OCL statement identifies a procedure member containing OCL and utility control statements placed into the job stream. If the procedure is not an MRT procedure, or if you did not specify PDATA=YES when the procedure member was created, the INCLUDE statement can pass *parameters* to the procedure.

If the procedure is a MRT procedure, or if you specified PDATA=YES when the procedure member was created, the INCLUDE statement can pass only *data* to a program.

For general information and programming considerations about MRT programs and procedures, see the *System/36 Environment Programming* book. For information about creating procedures, see Chapter 2, "Making Your Own Procedures."

## INCLUDE

Use the first form of the INCLUDE statement if the procedure name is the same as an OCL statement identifier. For example, if the procedure name is FILE, the following format is correct:

```
// INCLUDE FILE FILEA,FILEB
```

**Placement:** Place the INCLUDE statement anywhere among the OCL statements.

```
procedure name [ ,library name ] [ parm1,parm2...  
                                program data  
                                *ALL ]  
  
Or:  
  
// procedure name [ ,library name ] [ parm1,parm2...  
                                program data  
                                *ALL ]  
  
Or:  
  
// INCLUDE procedure name [ ,library name ] [ parm1,parm2...  
                                program data  
                                *ALL ]
```

59020307-0

### procedure name

This specifies the procedure member called. This procedure must exist in a source file named QS36PRC, which can be in one of the following:

- A specified library
- The current library
- The system library (#LIBRARY)
- The library list

### library name

This specifies the library searched for the procedure. If you specify a library name, the search is done in the following order:

1. The library specified
2. #LIBRARY
3. The library list

If you do not enter a library name, the search is done in the following order:

1. The current library
2. #LIBRARY
3. The library list

### parm1,parm2, ...

This specifies parameters for the procedure. You cannot use parameters if the procedure is an MRT procedure, or if you specified PDATA=YES when the procedure member was created. The parameters may or may not be required, depending on the procedure to which they are passed.

You must not exceed 128 characters in a parameter. You can pass a maximum of 64 parameters, separated by commas, with an INCLUDE statement. See "Procedure Parameters" on page 5-2 for more information on procedure parameters.

When entering parameters, you can type in up to 512 characters. For example, you can type in 32 sixteen-character parameters or 64 eight-character parameters. However, the combined total length of *all* param-

eters cannot exceed 1024 characters. You can accomplish this length by using substitution expressions and the local data area. See “Continuing the Lines of a Procedure” on page 2-7 for information about how to continue entering lines to get more than 120 characters.

#### **program data**

This specifies data, not parameters, passed at the first read operation in the program. The data starts with the first nonblank character following the procedure name and ends with the last nonblank character in the statement. The data passes to the program at the first input operation from the display station. (The input record in the program would contain this data.) Specify up to 508 characters using continuation. See “Continuing the Lines of a Procedure” on page 2-7 for information on how to continue entering lines for more than 120 characters.

**\*ALL** This specifies that all 64 parameters pass from the current procedure level to the procedure called. You can only specify \*ALL within a procedure. If you specify \*ALL as the only parameter and you enter it from the keyboard or a menu item selects it, an error message appears. If you specify \*ALL as one of the parameters, it is treated as a single parameter.

#### **Example 1**

In the following example, ACCTS and EMPNUM are two parameters interpreted by the PAYROLL procedure. Parameter 2 is omitted.

```
PAYROLL ACCTS, ,EMPNUM
```

#### **Example 2**

In the following example, MRTPROC is a MRT procedure that causes a MRT program to run. The number 126 is data passed to the MRT program on its first input operation from the requesting display station.

```
MRTPROC 126
```

---

## **INFOMSG OCL Statement**

The INFOMSG OCL statement specifies whether informational messages appear. For example, you might create a procedure that uses several system procedures. Informational messages are all messages that do not require a response. Most system procedures issue messages stating that they are running, and such messages might confuse some operators running application programs.

Also, an informational message sent to a remote display station results in longer response times because the current display is saved and then shown again when the procedure ends. By preventing the informational messages from appearing, you can decrease the time needed to run the procedure. When you specify the INFOMSG statement in a procedure, it remains in effect until the system processes another INFOMSG statement or until the procedure ends. When you enter the INFOMSG statement from the keyboard, it remains in effect until you enter another INFOMSG statement or until the operator signs off.

**Placement:** Place the INFOMSG statement anywhere among the OCL statements. You cannot use the INFOMSG OCL statement in an intersystem communications function (ICF) session, in a job running from the job queue, or in an evoked job.

```
// INFOMSG [ YES ]
             [ NO ]
```

**YES** This specifies that informational messages appear. If you do not specify a parameter, YES is assumed.

**NO** This specifies that informational messages do not appear.

### Example

The following example displays a general informational message, prevents the messages issued by the SAVE procedures from appearing, and then displays a message at the end of the procedure:

```
// * 'Saving files FILE1, FILE2, and FILE3'
// INFOMSG NO
SAVE FILE1,,VOL001
SAVE FILE2,,VOL001
SAVE FILE3,,VOL001
// INFOMSG YES
// * 'Save complete'
```

---

## JOBQ OCL Statement

The JOBQ OCL statement places a job on the job queue. The JOBQ OCL statement may be most useful for jobs that require no operator input or response, or for jobs whose output is not required by a subsequent job step. Enter up to 120 characters on the JOBQ statement.

Unexpected errors can occur if a job containing any of the following is placed on the job queue:

- Setting the system list service to the CRT (SYSLIST CRT), then running one of the following procedures or equivalent utility control statements:
  - CATALOG
  - LISTFILE
  - LISTDATA
- Running the LISTDATA procedure with CRT specified for the output parameter

When a job is placed on the job queue, the current files library is used as the initial session files library for the job on the job queue.

**Placement:** Place the JOBQ statement anywhere among the OCL statements.

```
// JOBQ [ job queue priority, ] [ library name ] ,procedure name
         [ 3, ] [ current library ]
         [ ,parm1,parm2... ]
```

### job queue priority

This specifies job queue priority (the job's order of processing from the job queue). The job queue priority is any number from 0 through 5.

When choosing the next job to run, the system considers jobs with higher priority numbers before jobs with lower priority numbers. For example, the system considers all jobs with a job queue priority of 5 before any other jobs in the job queue. The system considers jobs with the same job queue priority in the order in which they were placed in the job queue. Jobs with a job queue priority of 0 are the last jobs considered by the system. Job queue priority 0 is usually stopped. That is, any jobs placed on the job queue with a priority of 0 are not considered until the system operator starts priority 0.

Priority level 0 is not automatically stopped at initial program load (IPL) on the AS/400 system. To set up priority level 0 to run similar to priority level 0 on System/36, add the Change Job Queue Element (CHGJOBQE) command to your startup procedure (#STARTUP1 or #STARTUP2).

An example of the command is as follows:

```
CHGJOBQE SBSDB(QSYS/QBATCH) JOBQ(QGPL/QBATCH)
          MAXPTY8(0)
```

This example sets the maximum number of jobs active for System/36 environment priority level 0 to 0 and stops priority level 0 jobs from running.

If you do not specify a parameter, the system assigns a System/36 environment job queue priority based on the job priority (JOBPTY) attribute of the job description associated with the System/36 environment session that put the job on the job queue. Do not specify the comma shown in this parameter.

System/36 Environment Job Queue Priority	OS/400 Priority
5	3
4	4
3	5
2	6
1	7
0	8

**Note:** The JOBQ OCL statement submits jobs to the AS/400 job queue QBATCH in library QGPL.

#### **library name**

This specifies the library for the job. The system searches the specified library, then the system library, then the library list for the procedures, load members, message members, and display formats used in this job. If you do not specify a parameter, the current library is assumed.

#### **procedure name**

This specifies the procedure placed on the job queue.

#### **parm1,parm2 ...**

This specifies the parameters the procedure requires. Do not exceed 120 characters using the JOBQ OCL statement.

**Example**

The following example places the PAYROLL procedure, which is located in library PAYLIB, on the job queue with a job queue priority of 4:

```
// JOBQ 4,PAYLIB,PAYROLL
```

**LIBRARY OCL Statement**

The LIBRARY OCL statement specifies the name of the current library for the display station session or for the duration of a procedure. System/36 environment functions search the current library first for programs you specified on LOAD OCL statements, procedures, menus, message members, and display formats. If the System/36 environment function does not find the member in the current library, it automatically searches the system library (#LIBRARY), then the library list.

OS/400 functions do not search the libraries in the same order as the System/36 environment functions. The OS/400 system functions search the system libraries, then the current library, then the libraries in the user portion of the library search list. The library search list is under control of OS/400 system commands and may or may not contain #LIBRARY and QSSP.

The System/36 #LIBRARY now exists as two libraries: #LIBRARY and QSSP.

The library name parameter on the LOAD OCL statement can alter the search order for program and display formats. See "LOAD OCL Statement" on page 5-55.

A library remains current until another LIBRARY OCL statement is processed or until the display station session ends. If you use the LIBRARY statement in a procedure (and SESSION-YES is not specified), the current library is changed only while that procedure runs. When the procedure ends, the library that was current when the procedure began is again current.

When the operator signs on to the system or enters a LIBRARY OCL statement from the keyboard, a library is specified as current. That library remains allocated to the display station at least until you specify a different current library or NAME-0.

A program within a procedure can be interrupted by a system request. The sign-on that follows is to a new job. Neither the current library nor the session library of the interrupted job is used in the new job.

If you enter a LIBRARY OCL statement from a MENU display, one or more of the items on the displayed menu may not be correct because the procedures corresponding to those items do not exist in the new current library.

**Placement:** Place the LIBRARY OCL statement anywhere among the OCL statements except between a LOAD and RUN statement pair.

```
// LIBRARY NAME-0{library name} [ ,SESSION-NO{YES} ]
```

59020310-0

**NAME** This specifies the library that is the current library. If you specify NAME-0, the system only searches the system library (#LIBRARY).

**SESSION** This specifies changing the session library.

**NO** This causes change of the current library to the library specified in the NAME parameter. If you do not specify a parameter, NO is assumed. The change remains in effect until the current procedure ends or another LIBRARY OCL statement, which also changes the current library, is processed.

**YES** This causes change of the session library to the library specified in the NAME parameter (the session library is the library in effect at the keyboard). The change takes effect after the current job ends. It lasts until changed again or until the session ends.

**Note:** If the LIBRARY OCL statement is entered from the keyboard, the value specified for the SESSION parameter is ignored, and both the current and session values are changed.

### Example 1

The following example specifies the library MYLIB as the current library:

```
// LIBRARY NAME-MYLIB
```

### Example 2

The LIBRARY OCL statement in the following example specifies a library called PAYLIB as the current library for the procedure. The library contains the program named PAYROLL and the message member named PAYMSG.

```
// LIBRARY NAME-PAYLIB
// MEMBER USER1-PAYMSG
// LOAD PAYROLL
// FILE NAME-PAYFILE
// RUN
```

---

## LOAD OCL Statement

The LOAD OCL statement identifies a program to run. The LOAD statement is the first in a set of statements defining a job step.

The OS/400 program allows you to authorize programs with a security check that is part of the LOAD statement. If someone tries loading a damaged program, a locked program, or a program for which they are not authorized, the OS/400 program issues a message.

**Placement:** You cannot enter two LOAD OCL statements without an intervening RUN OCL statement.

```
// LOAD    program name [ ,library name ]
```

59020311-0

### program name

This specifies the program to load. The program named is a library load member.

## library name

This specifies the library in which to search for the program and display formats. If you specify a library name, the search is done in the following order:

1. The library specified
2. #LIBRARY
3. The library list

If you do not enter a library name, the search is done in the following order:

1. The current library
2. #LIBRARY
3. The library list

## Example 1

In the following LOAD statement, PAYROLL identifies a payroll program:

```
// LOAD PAYROLL
```

## Example 2

The following example is a sample procedure, stored in a library named MYLIB, that runs a program named MYPROG. The program displays messages from a message member named MESSAGES, which is stored in a library named COMMLIB.

```
// MEMBER USER1-MESSAGES, LIBRARY-COMMLIB  
// LOAD MYPROG, MYLIB  
// RUN
```

## Example 3

The following example is a procedure that runs a program named TEST. The program displays messages from a message member named TESTMSG. The message member and the program are both stored in a library named TESTLIB.

```
// LIBRARY NAME-TESTLIB  
// MEMBER USER1-TESTMSG  
// LOAD TEST  
// FILE NAME-TEMP, LABEL-TEMP?WS?, RETAIN-J, BLOCKS-30  
// RUN
```

---

## LOCAL OCL Statement

The LOCAL OCL statement modifies a specified area in the display station local data area. Use this statement to pass information between procedures, programs, job steps, and jobs.

A display station local data area exists for each command display station and each running MRT procedure. Each display station local data area is a 512-byte area on disk that you can use to pass information between jobs, job steps, procedures, and programs that run during a display station session. For information about using data from the display station local data area to modify a procedure, refer to “?L'position,length'? (Local Data Area) Expression” on page 3-19.

The OS/400 program automatically sets the local data area to blanks at the beginning of a session, when the operator signs on the system. All jobs run during the

session can use the local data area. However, a job placed on the job queue, or a released job step, uses a copy of the local data area as it existed when the job was placed on the queue, or when the job was released.

If the operator uses the inquiry mode by pressing the System Request key and selecting option 1, the contents of the local data area are saved and then restored when inquiry mode ends.

**Placement:** Place the LOCAL statement anywhere among the OCL statements.

```
// LOCAL [ OFFSET- $\left\{ \begin{array}{l} \text{position} \\ \hline \end{array} \right\} ] [ ,DATA-'characters' ] [ ,BLANK- $\left\{ \begin{array}{l} \text{length} \\ \hline *ALL \end{array} \right\} ]$$ 
```

```
[ ,AREA- $\left\{ \begin{array}{l} \text{USER} \\ \hline \text{SYSTEM} \end{array} \right\} ]$ 
```

59020312-0

Although each parameter is optional, you must specify either the DATA, BLANK, or AREA parameter.

**OFFSET** This specifies the first position in the display station local data area that changes.

**position** is any number from 1 through 512. If you do not enter a position, 1 is assumed.

**DATA** This specifies the data to place in the display station local data area.

**'characters'**

This value must be enclosed in apostrophes ('). If the characters contain embedded apostrophes (such as the apostrophe in o'clock), enter the embedded apostrophe as two apostrophes (for example: 'o'clock'). The number of characters is limited by the maximum length of the OCL statement. You can use double-byte character set (DBCS) characters.

**BLANK** This specifies the number of positions, starting with the position specified by OFFSET, to set to spaces.

**length** This specifies the number of positions to set to spaces.

**\*ALL** This specifies that all positions after the position specified by OFFSET are set to spaces.

If both BLANK and DATA are specified on the same LOCAL statement, the local data area is first set to spaces. Then the data is placed in the area.

**AREA** This specifies the area used by the current LOCAL statement, and by all following local data area substitution expressions.

**USER** This specifies use of the user local data area. If you do not specify AREA on the LOCAL statement, AREA-USER is assumed. The OS/400 and licensed programs do not affect this area.

**SYSTEM** This specifies use of the system local data area. The OS/400 program and other licensed programs use this area. *Any data* you place in this area may be lost and may affect the way the OS/400 program and other licensed programs

function. Therefore, you as a general user should not specify SYSTEM.

### Example 1

The following example places the word PAYROLL in the display station local data area starting in position 18 (through 24):

```
// LOCAL OFFSET-18,DATA-'PAYROLL'
```

### Example 2

The following example appears within a procedure and places the value of the first procedure parameter into the local data area, starting at position 12:

```
// LOCAL OFFSET-12,DATA-'?1?'
```

### Example 3

The following example clears positions 1 through 8 of the local data area, then places the procedure name in positions 1 through 8:

```
// LOCAL DATA-'?PROC?',BLANK-8
```

### Example 4

The following example sets the entire local data area to spaces:

```
// LOCAL BLANK-*ALL
```

---

## LOG OCL Statement

The LOG OCL statement indicates whether the OCL statements in a procedure are logged to the job log, regardless of the OCL statement logging indicator in the procedure. Therefore, you can create your procedures with this indicator set to off (for better performance) but still have the OCL statements logged to the job log when debugging the procedure is necessary.

If you want the System Help Support menu options and User menu options logged to the job log, specify // LOG ON. If you do not want the options logged, specify // LOG OFF or // LOG NORMAL.

All procedure commands are logged to the job log regardless of the LOG OCL statement setting and the procedure's logging attribute.

**Placement:** Place the LOG OCL statement anywhere among the OCL statements. When specified in a procedure, the LOG OCL statement remains in effect until the system processes another LOG statement or until the procedure ends. When you enter the LOG statement from the keyboard, it remains in effect until you enter another LOG OCL statement, the LOG procedure is entered, or the operator signs off.

```
// LOG { ON
        OFF
        NORMAL }
```

59020313-0

**ON** This specifies the logging of all OCL statements to the job log, regardless of the procedure's logging indicator. Also, menu options are logged to the job log.

**OFF** This specifies that no OCL statements are logged to the job log, regardless of the procedure's logging indicator. Menu options are not logged to the job log.

**NORMAL** This specifies use of the procedure's logging indicator. Menu options are not logged to the job log.

### Example 1

When you enter the following example from the keyboard, all processed OCL statements in the procedure are logged to the job log:

```
// LOG ON
```

### Example 2

The following example causes the logging of OCL statements only for procedures PROC1 and PROC3. The OCL statements for procedure PROC2 are not logged.

```
// LOG ON
PROC1
// LOG OFF
PROC2
// LOG ON
PROC3
```

---

## MEMBER OCL Statement

The MEMBER OCL statement specifies the names of the current message members used for procedures and programs. The system gets messages to display from the current message members. The specified members remain current until the system processes another MEMBER OCL statement from the display station, or until the display station session ends. If you use the MEMBER OCL statement in a procedure, the current members change only while the procedure runs. When the procedure ends, the message members current when the procedure began are again current.

The MEMBER statement specifies four types of message members: USER1, USER2, PROGRAM1, and PROGRAM2.

Your program's first-level and second-level messages are retrieved from the USER1 and USER2 members, respectively. First-level messages are up to 75 characters long. Second-level messages are up to 225 characters long.

For information about creating message members, see "CREATE Procedure" on page 4-66.

IBM licensed program first-level and second-level messages are retrieved from the PROGRAM1 and PROGRAM2 members, respectively. Your programs should not use the licensed program first-level and second-level message members.

**Placement:** Place the MEMBER OCL statement anywhere among the OCL statements.

## MEMBER

```
// MEMBER [ USER1-0{member name} ] [ ,USER2-0{member name} ]  
  
[ ,LIBRARY-library name ]  
  
[ ,PROGRAM1-0{member name} ] [ ,PROGRAM2-0{member name} ]
```

S9020314-0

**USER1** This specifies the name of the load member used for first-level messages for your programs and procedures. Use the USER1 message member with the // \* statement, the // \*\* statement, and procedure substitution expressions. If you specify 0 for member name, there is no current user first-level message member.

**USER2** This specifies the name of the load member used for second-level messages for your programs and procedures. If you specify 0 for member name, there is no current user second-level message member.

**LIBRARY** This specifies the library in which to search for the message members. If you specify a library name, the search is done in the following order:

1. The specified library
2. #LIBRARY
3. The library list

If you do not enter a library name, the search is done in the following order:

1. The current library
2. #LIBRARY
3. The library list

### PROGRAM1

This specifies the name of the load member used for IBM licensed program first-level messages. You need not specify this parameter in your procedures.

### PROGRAM2

This specifies the name of the load member used for IBM licensed program second-level messages. You need not specify this parameter in your procedures.

### Example 1

In the following example, the // \* statement causes a message to appear. The message with a message identification code of 0006 appears from the first-level message load member named MESSAGES.

```
// MEMBER USER1-MESSAGES  
// * 0006
```

### Example 2

The following example shows a sample procedure, stored in a library named MYLIB, that runs a program named MYPROG. The program displays messages from a message member named MESSAGES, which is stored in a library named COMMLIB.

```
// MEMBER USER1-MESSAGES,LIBRARY-COMMLIB
// LOAD MYPROG
// RUN
```

---

## MENU OCL Statement

The MENU OCL statement causes a specified menu to appear when the job containing the MENU OCL statement ends.

On the AS/400 system, you can also use the MENU OCL statement to display OS/400 menus (\*MENU objects) in addition to the System/36 environment menus. For more information on creating OS/400 menus, refer to the following:

- The *ADTS/400: Screen Design Aid for the System/36 Environment* book
- The Create Menu (CRTMENU) CL command in the *CL Reference* book
- The *System/36 Environment Programming* book.

The menu can contain double-byte characters. If you attempt to display a DBCS menu at a non-DBCS display station, the OS/400 program issues an error message.

**Placement:** Place the MENU statement anywhere among the OCL statements. The MENU OCL statement is ignored if it occurs in a MRT procedure.

```
// MENU    menu name [,library name ]
```

59020315-0

### menu name

This specifies the menu to display at the end of the job, and can contain from 1 to 6 characters.

### library name

This specifies the library containing the MENU. If you specify a library name, the search is done in the following order:

1. The specified library
2. #LIBRARY
3. The library list

If you do not specify a library name, the system does the search for the menu in the following order:

1. The current library
2. #LIBRARY
3. The library list

The specified library becomes the session library. If the MENU statement is within a procedure, the current library for the procedure is not affected.

## Example 1

In the following example, the MENU OCL statement causes a menu called DAILY, which resides in the current library, to appear when the job containing the MENU statement ends:

```
// MENU DAILY
```

## Example 2

In the following example, the MENU OCL statement causes a menu called PAYROL, which resides in a library named PAYLIB, to appear when the job containing the MENU OCL statement ends. PAYLIB becomes the current library and the session library after the job ends.

```
// MENU PAYROL,PAYLIB
```

---

## MSG OCL Statement

The MSG OCL statement sends a message to any of the following:

- A selected display station
- A selected user
- All display stations
- A selected personal computer location
- The system operator
- A user on another system

Messages sent are placed in one of the following types of OS/400 message queues:

- Each display station has an associated work station message queue. The name of this message queue is the same as the OS/400 device name, not the System/36 environment display ID.
- Each user on the system has an associated user profile message queue.
- Each personal computer location has an associated message queue. The name of this message queue is the same as the personal computer location name.
- The system operator message queue is QSYSOPR.

Messages sent to a message queue are retained on the queue until deleted by a user.

**Note:** This is different from the System/36, on which all user messages displayed or older than 7 days at IPL were deleted, and all work station messages displayed or at IPL were deleted.

The message waiting light is set on when a message is sent to a message queue that is in notify delivery mode and the severity of the message is greater than the severity code filter for the message queue. Once messages are displayed, the message waiting light will be reset.

**Note:** Messages sent using the MSG OCL statement use the following severities:

- 00 for (user id,address), (list id,qualifier), or group name
- 99 for ALL
- 80 for all other messages

The following search order is used to find the message queue to which the message is to be sent:

1. (user id,address), or (list id,qualifier)
2. ALL
3. blank
4. System/36 environment display ID
5. User ID
6. OS/400 display station, personal computer location, or user-defined

The message sent will be placed in the first message queue in your library list that matches the specified name.

7. group name

If a user is not signed on when a message is sent, and the user has a user profile, the message is placed in the user profile message queue. The message can then be displayed by the user when he signs on.

See the *System Operation* book for more information on message queues. See the *System/36 Environment Programming* book for more information on messages in the System/36 environment.

**Placement:** The MSG OCL statement can appear anywhere among the OCL statements.

The MSG OCL statement can be used in the #STRTUP1 or #STRTUP2 procedure.

```
// MSG [ &sysse. display id
        &pcpf. display id
        user id
        pc location
        user-defined
        (user id, address)
        (list id, qualifier)
        ALL
        group name ] ,message text
```

59020676-0

If the first parameter is not specified and the second parameter is specified, the message is sent to the system operator message queue (QSYSOPR) in library QSYS.

### System/36 environment display id

This specifies the System/36 environment 2-character display ID to which the message is to be sent to. The message is sent to the work station message queue of the associated OS/400 display station. To find out the System/36 environment display ID and its associated OS/400 display station use the Display System/36 (DSPS36) command.

### OS/400 display station

This specifies the 1- to 8-character OS/400 display station to which the message is to be sent to.

**user id** This specifies the 1- to 8-character user profile name that identifies the user profile message queue of the user to whom the message is to be sent. If you specify QSYSOPR, the message will be sent to the user profile message queue associated with the QSYSOPR user profile.

**pc location**

This specifies the 1- to 8-character personal computer location name to which the message is to be sent.

**user-defined**

This specifies the 1- to 8-character name of a user-defined message queue to which the message is to be sent. The user-defined message queue must be in a library in your library list.

**(user id,address)**

This specifies the 1- to 8-character user ID and the 1- to 8-character address that together identify the person in the distribution network to whom the message is to be sent to. The user ID and address must already exist in the system distribution directory. See the *SNA Distribution Services* for more information on the system distribution directory.

**(list id,qualifier)**

This specifies the 1- to 8-character distribution list ID and the 1- to 8-character distribution list ID qualifier that together identify a list of users enrolled in the system distribution directory for SNA Distributions Services to whom the message is to be sent. See the *SNA Distribution Services* for more information on the system distribution directory.

**ALL**

This specifies that the message is sent to all display stations and personal computer locations on the system. Messages sent to all display stations are always sent in break mode.

**group name**

This specifies the 1- to 8-character name of a distribution list ID. The System/36 environment will use your system name as the distribution list ID qualifier. The distribution list ID and the distribution list ID qualifier together identify a list of users enrolled in the system distribution directory for SNA Distributions Services to whom the message is to be sent. See the *SNA Distribution Services* book for more information on the system distribution directory.

**,message text**

This specifies the contents of the message to be sent. Up to 75 characters can be entered. DBCS messages are always sent, but may only display correctly on DBCS capable display stations.

**Example**

In the following example, a procedure is run from the job queue or evoked, and the operator wants to know when the procedure ends. The MSG OCL statement at the end of the procedure informs the display station operator when the procedure ends.

```
// LOAD PROGRAM1
// FILE NAME-FILE1
// RUN
// MSG ?WS?,PROGRAM1 IS COMPLETE
```

The ?WS? expression causes the work station ID of the display station that placed PROC A on the job queue to be substituted into the MSG statement. For information about substitution expressions, see "Substitution Expressions" on page 3-7.

## NOHALT OCL Statement

The NOHALT OCL statement specifies the automatic response severity level for the system, a job, or a job step. This specification allows the system to respond to messages that have automatic response values, rather than requiring that an operator enter an error message response. This is helpful when you run the system without an operator (overnight, for example).

For more information on automatic response, see “RESPONSE Procedure” on page 4-196.

**Placement:** Place the NOHALT OCL statement anywhere among the OCL statements.

If you enter NOHALT from the keyboard and do not specify the second parameter, it remains in effect until you enter another NOHALT OCL statement or sign off the system. If a job specifies NOHALT and you do not specify the second parameter, it remains in effect until another NOHALT OCL statement processes, or until the job ends.

```
// NOHALT severity level, [JOB
                           SESSION
                           SYSTEM]
```

59020317-0

When setting the automatic response level for MRT procedures, it is necessary to include a // NOHALT X,JOB, where X is the severity level required, in the MRT procedure. This will insure a consistent severity level for each user who attaches to the MRT program. If a // NOHALT X,JOB statement is not included in the MRT procedure, the severity level is set to 0, and will remain at 0 until the procedure ends.

### severity level

This specifies the automatic response severity level. Enter 0, 1, 2, 3, or 4.

- 0** This specifies that no messages receive an automatic response from the system. If a message appears, the operator must enter a response to the message. When you specify 0, the automatic response is turned off.
- 1** This specifies that any messages having a severity level of 1 receive an automatic response from the system.
- 2** This specifies that any messages having a severity level of 1 or 2 receive an automatic response from the system.
- 3** This specifies that any messages having a severity level of 1, 2, or 3 receive an automatic response from the system.
- 4** This specifies that any messages having a severity level of 1, 2, 3, or 4 receive an automatic response from the system.

### JOB

This specifies the severity level used for the job. The specified level remains in effect until the job ends, or until the level is changed by another NOHALT OCL statement or a NOHALT procedure.

## OFF

If the NOHALT statement is within a procedure and you do not specify a second parameter, JOB is assumed. If you enter the NOHALT statement at the keyboard and specify JOB, the second parameter is ignored, and the NOHALT status is set for the session.

### SESSION

This specifies the severity level used for the display station session. The specified level remains in effect until the display station session ends, or until the level is changed by another NOHALT OCL statement or a NOHALT procedure.

If the NOHALT statement is within a procedure and you specify SESSION, the specified level does not take effect until the current job ends. If you enter the NOHALT statement from the keyboard and do not specify a second parameter, SESSION is assumed.

**SYSTEM** This specifies whether the severity level establishes the system automatic response severity level and the severity level for the display station session. The system severity level establishes the severity level for other display stations that sign on *after* the system processes the SYSTEM parameter. Other display stations that were already signed on are not affected.

To establish the system severity level, include the NOHALT statement in the IPL start-up procedure #STRTUP1. See “#STRTUP1 Procedure” on page 4-3 for more information.

If the NOHALT statement is within a procedure and you specify SYSTEM, the specified level takes effect immediately for the system automatic response severity level, but does not take effect for the display station session until the current job ends.

An operator must have \*JOBCTL special authority to specify SYSTEM. Refer to the Change User Profile (CHGUSRPRF) command in the *CL Reference* book for more information on special authorities.

### Example 1

To establish an automatic response severity level of 3 for the entire system, create a procedure named #STRTUP1 that contains the following statement:

```
// NOHALT 3,SYSTEM
```

### Example 2

To establish an automatic response severity level of 3 for the payroll program, specify the following statement in the payroll procedure:

```
// NOHALT 3  
// LOAD PAYROLL  
// FILE NAME-EMPLOY,DISP-OLD  
// RUN
```

---

## OFF OCL Statement

The OFF OCL statement immediately signs an operator off the system, thus ending a display station session. After the session ends, the sign-on display appears at the display station.

You cannot use the OFF OCL statement in a MRT procedure. Also, you cannot use it in an ICF session, in a job running from the job queue, or in a job that was evoked.

The OFF OCL statement is mapped to the SIGNOFF CL command. If the command default for the SIGNOFF CL command on your system is LOG(\*NOLIST), a job log is not produced for the job when the OFF OCL statement is processed. If the command default for the SIGNOFF CL command on your system is LOG(\*LIST), a job log is produced for the job when the OFF OCL statement is processed. The command defaults for the SIGNOFF CL command are changed using the Change Command Default (CHGCMDDFT) CL command. See the *CL Reference* book for more information on the SIGNOFF and CHGCMDDFT CL commands.

**Placement:** Place the OFF OCL statement anywhere among the OCL statements except between the LOAD and the RUN OCL statements.

```
// OFF      [ DROP  
              HOLD ]
```

59020318-0

Use DROP and HOLD only for remote display stations on switched communications lines. These parameters are ignored for other display stations.

**DROP** This specifies that the communications session for this display is no longer needed.

**HOLD** This specifies holding of the communications session for the display. A sign-on display appears.

**Note:** On System/36, DROP was the default. The OS/400 default is to use the attribute defined in the device description for the display station. Use the Change Device Description (Display) (CRTDEV DSP) command specifying \*YES for the drop line at sign-off (DROP) parameter.

### Example

In the following example, a procedure runs the program LAST from a display station and, at the end of the job, signs the user off the system:

```
// LOAD LAST  
// RUN  
*  
// OFF
```

---

## POWER OCL Statement

Not supported. Use the Power Down System (PWRDWNSYS) command to power off the system.

**Note:** The PWRDWNSYS command does not check for active jobs before powering down the system.

---

## PRINTER OCL Statement

The PRINTER OCL statement identifies and controls the output for a printer file created by one or more job steps. Use the PRINTER OCL statement to specify the following printer file changes:

- The specific output queue to which spooled printed output is directed
- Whether the output is printed on one side or two sides of the paper
- The horizontal print density (characters per inch)
- The forms number used
- The number of copies of spooled output printed
- The spooled output priority
- The amount of space left for the shift-out and shift-in characters (DBCS only)
- The identification number of the print wheel to use, or the identification number of the font to use
- The printer drawer from which to use paper
- The orientation or size of the printed output on the page (rotation or reduction)
- If printing of spooled output begins before the job step completes
- If the spooled file holds printed output after it prints
- If the operator lines up the forms in the printer before printing begins
- If the printer file continues with following job steps
- If output prints on a double-byte-character printer
- If the output is right-justified
- If the output prints in final quality
- If the end-of-forms message appears
- Whether to spool printed output
- Whether to process any extended characters printed (DBCS)

**Placement:** Place the PRINTER OCL statement anywhere among the OCL statements. You can specify a PRINTER OCL statement for each printer file the job step uses. If you do not use a PRINTER OCL statement, the System/36 environment uses the PRINTER statement default values when printing the output. The defaults can come from the following:

- System configuration
- PRINT procedure
- LINES procedure
- SET procedure
- FORMS OCL statement

```

// PRINTER NAME- { file name } [ ,DEVICE- { printer id } ]
                { $SYSLIST } [ SYSTEM ]

[ .OUTQ- { output queue name } ]
  { library name/output queue name }

[ .DUPLEX- { NO } ]
  { YES }
  { TUMBLE }

[ .LINES- lines per page ] [ .LPI- { 4 } ] [ .CPI- { 10 } ]
  { 6 }
  { 8 }
  { 15 }

[ .FORMSNO- forms number ] [ .ALIGN- { YES } ] [ .SPOOL- { YES } ]
  { NO }
  { NO }

[ .COPIES- { number } ] [ .CONTINUE- { YES } ] [ .PRIORITY- { 1 } ]
  { 1 }
  { NO }
  { 5 }
  { 4 }
  { 3 }
  { 2 }
  { 0 }

[ .ACTIVITY- { NORMAL } ] [ .DEFER- { YES } ] [ .HOLD- { NO } ]
  { HIGH }
  { MEDIUM }
  { LOW }
  { NO }
  { YES }

[ .IGCCPI- { 5 } ] [ .SOSI- { NORMAL } ] [ .TYPE- { ANY } ]
  { 6.7 }
  { SHIFT }
  { DROP }
  { IGC }
  { IGC18 }
  { IGC24 }

[ .EXTN- { ON } ] [ .JUSTIFY- { 100 } ] [ .FONT- { id } ]
  { OFF }
  { 50 }
  { 0 }

[ .TEXT- { YES } ] [ .EOPMSG- { YES } ] [ .ROTATE- { 0 } ] [ .DRAWER- { 1 } ]
  { NO }
  { NO }
  { 90 }
  { 180 }
  { 270 }
  { COR }
  { 2 }
  { 3 }

```

813P001.2

**NAME** This specifies the name the program uses to refer to the printer file. Use any combination of characters (numeric, alphabetic, and special). Use commas (,), periods (.), apostrophes ('), blanks, question marks (?), slashes (/), hyphens (-), greater than signs (>), plus signs (+), and equal signs (=) with caution because they have special meanings within procedures. The first character of a file name must be alphabetic (A through Z, #, \$, or @). Do not exceed 8 characters.

When you do not place the PRINTER OCL statement between a LOAD and RUN OCL statement pair, and do not specify the CONTINUE parameter, you cannot specify the NAME parameter.

**\$SYSLIST**

This specifies that the PRINTER statement controls system list output. For information about system list output, see "SYSLIST OCL Statement" on page 5-98.

If the NAME parameter does not match a file name specified in the program, the System/36 environment uses the default values of the PRINTER statement when printing the output.

**Note:** \$SYSLIST is ignored if you turn on debugging. For more information, refer to “DEBUG OCL Statement” on page 5-22.

**DEVICE** This specifies the printer used for the print file.

**printer id** This specifies the work station ID of the printer used. You can use the Display System 36 (DSPS36) command to determine the printer IDs.

**SYSTEM** This specifies usage of the system printer.

If you do not specify DEVICE, the following conditions apply:

- If you specify NAME-\$SYSLIST, the PRINTER OCL statement overrides the system list device setting. For example, if the current system list device is CRT, the output prints rather than appears.
- If you specify a file name other than \$SYSLIST, and the procedure is a MRT procedure, the printer output goes to the system printer.
- If the job step containing the PRINTER statement is released, evoked, or placed on the job queue, the printed output goes to the printer specified during system configuration. The output goes either to the session printer or to the system printer.
- If the job step runs as a SRT procedure, the printer output goes to the session printer.

**OUTQ** This specifies the name of the output queue to which spooled printed output is directed. You can specify the name of the output queue alone. Or, you can specify the name of the library containing the output queue with the name of the output queue. If both the library name and the output queue name are specified, they must be separated by a slash (/). If a library name is not specified, the output queue is located through the library list of your OS/400 job. The names of output queues and libraries can each be up to 10 characters in length. If the OUTQ parameter is specified along with the SPOOL-NO parameter, the value for the SPOOL parameter is ignored.

**DUPLEX** This specifies how the printed information appears on a page.

**NO** Produces printed information on one side of a sheet of paper.

**YES** Prints on both sides of a sheet of paper. The information is correctly formatted if the paper is bound on the long edge of the paper. This option should only be specified for printers capable of duplex printing.

**TUMBLE** Prints on both sides of a sheet of paper. The information is correctly formatted if the paper is bound on the short edge of the paper. This option should only be specified for printers capable of tumble duplex printing.

If you do not specify DUPLEX, NO is assumed.

**LINES** This specifies the number of print lines per page. The number of lines per page is any number from 1 through 112. If you do not specify LINES, the number of lines set previously by a FORMS OCL statement, LINES procedure, or PRINT procedure is used. If you specify the number of lines per page in a program, the program's value is used. If the number of lines per page was not set during the session, the value

specified for the display station during system configuration or assigned by the SET procedure or the \$SETCF utility program is used.

For most user-written programs and System/36 environment utility programs that use the system list device for output, the following is true:

- Printing starts on line 6 of the first page and on line 6 of all new pages.
- If the value specified for LINES is greater than 12, printing skips to a new page when six less than the number of lines specified is printed.

For example, if you specify LINES-66, printing starts on line 6 and skips to a new page after printing line 60. Therefore, you get 55 lines of output per page (66 – 11 = 55, 5 blank lines at the top of a page and 6 blank lines at the bottom). If you specify LINES-13, two lines are printed per page. When you specify 12 or fewer lines per page, printing occurs on every line of the page.

For System/36 environment utility programs that use OS/400 print files for output, the following is true:

- Printing starts on line 1 of the first page and on line 1 of all new pages.
- If the value specified for LINES is greater than 12, printing skips to a new page when six less than the number of lines specified is printed.

For example, if you specify LINES-66, printing starts on line 1 and skips to a new page after printing line 60. Therefore, you get 60 lines of output per page (66 – 6 = 60, 6 blank lines at the bottom of the page). If you specify LINES-13, seven lines are printed per page. When you specify 12 or fewer lines per page, printing occurs on every line of the page.

**Note:** For a list of the System/36 environment utilities that use the system list device and the OS/400 print files, see the *System/36 Environment Programming* book.

For print operations from your programs, the System/36 environment indicates a printer overflow condition when six less than the number of lines specified print (unless the program uses another value).

Some printers do not allow certain values for lines per page. Refer to your printer documentation for more information.

**LPI** Lines per inch specifies the vertical print density to use for printed output from the display station session.

If you do not specify LPI, the system uses the LPI value set previously by a FORMS OCL statement, LINES procedure, or PRINT procedure. If you do not specify the LPI parameter and the LPI value was not previously set during the session, the system uses the value specified when the printer was configured. Because output can be directed to any printer from the spool file, this default configuration value is determined when the output is printed, not when it is intercepted in the spool file.

## PRINTER

**CPI** Characters per inch specifies the horizontal print density the printer uses for output. You can specify 10 or 15.

If you do not specify a CPI value, the cpi density set previously by a FORMS statement, LINES procedure, or PRINT procedure is used. If you do not specify a CPI value and the characters per inch has not been previously set during the session, the system uses the value specified when the printer was configured (either 10 or 15).

See also FONT later in the description of this OCL statement.

### FORMSNO

This specifies the form number of the printer form used for printed output from the display station session. (Each type of form should have a unique user-assigned form number.) The form number can be any combination of up to 4 characters except commas (,), apostrophes ('), asterisks (\*), and blanks.

#### Notes:

1. Use question marks (?), slashes (/), equal signs (=), greater than signs (>), plus signs (+), and hyphens (-) with caution because they have special meanings within procedures.
2. FORMSNO is reinitialized when the printer is varied off. This causes the form type message to be sent to the printer the first time anything is printed.

If you do not specify FORMSNO, the system uses the forms number previously set by a FORMS statement. If the forms number was not set during the session, the system uses the forms number specified for the display station during system configuration, or the forms number assigned by the SET procedure or \$SETCF utility program.

### ALIGN

This allows the operator controlling the printer to line up the forms in the printer before the printing of output for the job step begins. If you specify the ALIGN parameter, the system ignores any alignment indicator used in the program. If you do not specify ALIGN, the system uses the alignment indicator specified in the program (if any).

**YES** This specifies that the system prints the first line of output and displays a message. The operator can then line up the forms and then either:

- Select an option causing a line to print again and the alignment message to appear again.
- Select an option causing printing to continue with the rest of the output.

**NO** This specifies that the system prints the output without allowing the line-up of the forms.

### SPOOL

This specifies whether the printer output is spooled.

**YES** This specifies the spooling of output.

**NO** This specifies that output is not spooled, and that the printer used in the DEVICE parameter is assigned to the job or job step. If the printer is not available, a message appears.

If spooling was selected during system configuration, SPOOL-YES is assumed. Otherwise, the system ignores SPOOL.

**COPIES** This specifies the number of copies of spooled printer output printed for the job step. Specify any number from 1 to 255. If you do not specify COPIES, 1 is assumed. The COPIES parameter affects only spooled output.

**CONTINUE**

This specifies whether the printed output continues, that is, whether the printout from this job step and from following job steps are considered as a single print step. You can only specify the CONTINUE and NAME parameters together when you place the PRINTER statement between the LOAD and RUN OCL statements.

**Note:** If the printer output has been spooled, the spool file entry processes as if it were created by one job step. If you specify different print file characteristics (for example, a different number of lines per page) in steps within the job, those values can be used only when the spool file entry prints from beginning to end.

**YES** This specifies that a specific printer continues printing in following job steps. The DEVICE parameter specifies the printer whose output continues.

Printing continues until another PRINTER statement containing CONTINUE-NO processes for that device, or until the job ends. This continuous printing allows you to group the output of several job steps together, rather than possibly having each job step's output interspersed with the output from other jobs on the system.

**Note:** A page eject occurs between the output of each job step.

If the printer output is being spooled, the spool file entry associated with the printer output is left open. If the printer output is not being spooled. That is, the output is being sent directly to a printer, the printer remains allocated to the job.

If two or more printer files in the same job step refer to the same printer, only one of the printer files from the job step continues. If print spooling is used, the other printer files are placed on the spool file as separate entries. The following specifications determine which printer file continues:

- If you specify a PRINTER OCL statement for the same printer device, the system continues the print step represented by the first PRINTER OCL statement.
- If you do not specify a PRINTER OCL statement for the same printer device, the system continues the first print step allocated by the program that uses the printer device.

When a print file is continued, all parameters are ignored for that device except NAME, DEVICE, and CONTINUE.

If you release a job step by using the RELEASE parameter on the ATTR OCL statement, or a job step runs a MRT program, those steps cannot use continued print files defined by other steps of the job. Those steps must have their own PRINTER OCL statement for that print file.

## PRINTER

**NO** This specifies that the print step no longer continues. The following specifications determine when the print step really closes:

- If you do not place the PRINTER statement between a LOAD and RUN OCL statement pair, the continued print step is immediately considered complete.
- If you place the PRINTER statement between a LOAD and RUN OCL statement pair, the continued print step completes after the current job step ends (or after the program closes the printer file).

When the print step completes, the spool entry is indicated as complete or the printer is released from the job.

### PRIORITY

This specifies the priority of spooled output from the job step. The priority is any decimal number from 5 to 0. The system prints spool file entries with higher priority numbers before jobs with lower priority numbers. For example, all printed output with a priority of 5 prints before any other spool file entries. Spool file entries with the same priority print in the order in which they were placed on the spool file. Priority 1 entries are the last entries printed by the system. If you do not specify the PRIORITY parameter, PRIORITY-1 is assumed.

The system places priority 0 entries on the spool file with a priority of 1 and holds them. These entries do not print until a RELEASE control command is entered to specifically cause them to print.

The actual order of the printed output may change because of the forms number used. See the PRT parameter in "START Control Command" on page 6-22 for how the order can be changed.

### ACTIVITY

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used.

**DEFER** This specifies whether the system can begin printing spooled output before the print step completes. The DEFER parameter affects only spooled output.

**YES** This specifies that the system should not print spooled output from the job step until the print step completes. If you do not specify DEFER, DEFER-YES is assumed.

**NO** This specifies that the system can begin printing spooled output from the print step before the step completes.

If you specify DEFER-NO for material that takes a long time to print, you might reduce the total time needed to run the program and obtain the printout.

**HOLD** This specifies whether to hold the printed output on the spool file after it prints. To hold printed output in the spool file without it printing, see the PRIORITY parameter.

**NO** This specifies that the printed output is not held and is removed from the spool file after it prints. If you do not specify HOLD, HOLD-NO is assumed.

- YES** This specifies that the printed output is held after all copies print. The number of copies held is set to one.
- IGCCPI** Double-byte characters per inch specifies the horizontal print density to use for DBCS printed output. You can specify 5 or 6.7. If you do not specify IGCCPI, IGCCPI-5 is assumed.
- SOSI** Shift-out shift-in specifies the space left for the shift-out (hex 0E) and shift-in (hex 0F) characters when printing.
- NORMAL** This specifies that a space is substituted for each shift-out and shift-in character for printing. If you do not specify the SOSI keyword, NORMAL is assumed.
- SHIFT** This specifies that no space is left for the shift-out (hex 0E) character and that two spaces are left for the shift-in (hex 0F) character for printing.
- DROP** This specifies that no spaces are left for the shift-out or shift-in characters. During printing, all characters are moved to the left to fill the spaces deleted for the shift-out and shift-in characters.
- TYPE** This specifies on which printer the output prints.
- ANY** This specifies that the printer output can print on any type of printer. If the printer encounters a DBCS character and cannot print DBCS characters, the character position is filled with blanks. If you do not specify the TYPE parameter, TYPE-ANY is assumed.
- IGC, IGC18, IGC24**  
This specifies that the output contains DBCS characters and should be printed on a printer that prints DBCS characters. If the printer cannot print DBCS characters and if the output is not spooled, DBCS characters are deleted. If the printer cannot print double-byte character set characters and if the output is spooled, the operator can choose to hold the spooled output or to print it, with blanks substituted for DBCS characters.
- Note:** If a system does not have the double-byte character set version of the OS/400 program, the printing of DBCS characters is unpredictable.
- EXTN** This specifies whether extended characters in the printer output are to be processed. The EXTN parameter is for the DBCS version of the OS/400 program and is ignored for non-DBCS systems. If you specify NAME-\$SYSLIST and the system list device is a printer, the EXTN parameter overrides the value specified on the SYSLIST OCL statement.
- ON** This specifies that extended characters in the printer output are processed. If you do not specify the EXTN parameter, ON is assumed.
- OFF** This specifies that the extended characters are not processed. The system-defined default double-byte character prints for any extended characters in the output.

## PRINTER

- JUSTIFY** This specifies the percentage that output from a printer right-justifies. The system sets the right margin for justification based on the record length. For example, if the record length is 80, all text is justified to column 80. You can control justification with a program by using skipping and spacing values. If you specify skipping and spacing values before printing, the printed lines are justified. If you specify skipping and spacing values after printing, the printed lines are not justified. If the output is printed on a printer that does not support right margin justification, the system ignores the JUSTIFY parameter.
- 100** This specifies that the printed lines are evenly adjusted at the right margin (right-justified).
- 50** This specifies that the printed lines are right-justified to a position halfway between the end of the text and the right margin (that is, ragged right edge).
- 0** This specifies that the printed lines are not right-justified. If you do not specify the JUSTIFY parameter, 0 is assumed.
- FONT** This specifies the identification number of the print wheel or of the font to print. All valid AS/400 fonts in the range of 0 through 255 are supported. The value must be specified as a 1-byte hexadecimal number. See the *Printer Device Programming* book for more information on fonts and printers.
- If you specify a value for both the CPI parameter and the FONT parameter, an error message appears.
- TEXT** This specifies whether the output prints in draft or final quality.
- YES** This specifies that the printout is in final quality.
- NO** This specifies that the TEXT keyword is not used on the PRINTER OCL statement. The print quality set on the printer operator control panel is used.
- EOFMSG** This specifies whether the statement issues the end-of-forms message to the console or subconsole operator.
- NO** This specifies that the statement does not issue the end-of-forms message.
- YES** This specifies that the statement issues the end-of-forms message. If you do not specify the EOFMSG parameter, YES is assumed.
- ROTATE** This specifies that the output is rotated on the page or that the size of the printed output is reduced. This parameter is valid only if output is printed on a printer that supports rotation of printed output.
- 0** This specifies that printed output rotates zero degrees on the page.
- 90** This specifies that printed output rotates 90 degrees on the page.
- 180** This specifies that printed output rotates 180 degrees on the page.
- 270** This specifies that printed output rotates 270 degrees on the page.

**COR** Computer output reduction specifies that the size of the printed output reduces. Output normally printed on 14-inch wide paper prints on 8-1/2 x 11-inch paper.

**DRAWER** This specifies that the paper a printer uses is selected from the specified drawer.

- 1** This specifies that output is printed on paper from drawer 1. If you do not specify the DRAWER parameter, 1 is assumed.
- 2** This specifies that output is printed on paper from drawer 2.
- 3** This specifies that output is printed on paper from drawer 3.

See the *CL Reference* book for information about the CRTPTF command, and for valid combinations of these parameters for available printers.

### Example 1

The following example shows that for the program named PROGRAM1, output to the file called PRINT1 goes to the system printer and is spooled. Three copies print, and the OS/400 program can begin printing before the job step ends.

```
// LOAD PROGRAM1
// PRINTER NAME-PRINT1,COPIES-3,DEFER-NO,
//      DEVICE-SYSTEM
// RUN
```

### Example 2

The following example shows that the output from the three programs prints as a single report on the session printer:

```
// LOAD PROG1
// PRINTER NAME-PRINT1A,CONTINUE-YES
// RUN
// LOAD PROG2
// PRINTER NAME-PRINT1B
// RUN
// LOAD PROG3
// PRINTER NAME-PRINT1C
// RUN
```

### Example 3

The following example shows that for the program named PROGRAM2, output to the file called PRINT2 is spooled in the output queue named OUTQ2 in the library named PAYROLL:

```
// LOAD PROGRAM2
// PRINTER NAME-PRINT2,OUTQ-PAYROLL/OUTQ2
// RUN
```

---

## PROMPT OCL Statement

The PROMPT OCL statement allows you to:

- Prompt for up to 64 procedure parameters by using one or more display formats. You can prompt for up to 1024 characters .
- Define each parameter for the operator.
- Assign default parameters.

## PROMPT

- Show the display format to read on the first read operation to the display station in a program. This format is also called read-under format (RUF).
- Control various display format functions.

The PROMPT OCL statement provides a return code (returns a value to the procedure) indicating which command key or function key was pressed. You can use the ?CD? substitution expression to determine the value of this code. See “?CD? (Return Code) Expression” on page 3-13 for more information. The following chart shows the command and function keys and their return codes. These return codes are set only if you do not specify PDATA=YES.

Return Code	Key Pressed
0000	Enter/Rec Adv
2001 through 2024	Command keys 1 through 24
2090	Page Down
2091	Page Up
2092	Help
2093	Record Backspace (Press the Home key while the cursor is in the Home position)

See the *System/36 Environment Programming* book for an explanation of how to design and enter display formats. You cannot use the PROMPT OCL statement to show a display format that contains more than 1024 characters of run-time output data.

The following rules apply to the SFGR D-specifications for input fields and input/output fields on a format displayed by the PROMPT statement:

- You must define fields in order by parameter number on the D-specification. Fields do not, however, have to appear in parameter-number order on the display screen. Line number and horizontal position entries determine the position on the display screen.
- Normally, the output-data entry (columns 23 and 24) should be an indicator number that corresponds to the parameter position in the procedure. If a parameter has already been assigned a value when the prompt format appears, the indicator for the parameter is turned on, and the assigned value appears. If the parameter has not already been assigned a value, the constant default value coded in columns 57 through 64 appears.
- You must specify Y in the input-allowed entry (column 26).
- You can code a default value in the constant-data entry (columns 57 through 64).

**Placement:** Place the PROMPT OCL statement anywhere among the OCL statements. It cannot be entered at the keyboard, and is not allowed from the job queue or from a job started by the EVOKE OCL statement.

```
// PROMPT MEMBER-display format load member,FORMAT-display format name

[ ,LIBRARY-library name ] [ ,START-1{parameter number} ]

[ ,LENGTH-n 'n,n...' ] [ ,PDATA-NO{YES} ] [ ,UPSI-NO{YES} ]
```

59020321-0

**MEMBER** This specifies the name of the display format load member that contains the display format.

**FORMAT** This specifies the name of the display format that appears.

**LIBRARY** This specifies the library in which to search for the display format load member. If you specify a library name, the search is done in the following order:

1. The specified library
2. The system library (#LIBRARY)
3. The library list

If you do not enter a library name, the search is done in the following order:

1. The current library
2. #LIBRARY
3. The library list

**START** This specifies the number of the first procedure parameter to display and prompt for. Any previously defined parameters whose positions are less than the position specified are not affected.

**parameter number**

This is any number from 1 to 64. If you do not specify a parameter, the first parameter is assumed.

**LENGTH** This specifies the number of display format input and output positions used by each procedure parameter being displayed and prompted for.

**n** This specifies the length of one parameter (quotation marks are not necessary) and the value applies to the parameter specified by START.

**'n,n...'** This specifies the length of more than one parameter, starting with the parameter that START specifies. Each **n** can have a value of 0 (zero) through 128. The total length cannot exceed 1024 characters.

If you do not specify a value for a position (for example: ',,'), the parameter has a length of 8 characters.

A value of 0 (zero) indicates that the parameter is not displayed or prompted for, and the input field for that parameter is skipped. If you specify 0 for the parameter's length, the parameter's value and length do not change from what they were before the PROMPT statement processed.

When the display format is shown, the parameters appear beginning in the farthest left position in the display format field. If the actual length of the parameter displayed is greater than the corresponding length value

## PROMPT

specified, the displayed parameter's farthest right characters are truncated (only the farthest left number of characters specified appear).

After the operator types in the parameters and presses the Enter key or a valid command key, the length of each parameter is determined by counting the number of characters entered from left to right. All characters, including leading and embedded blanks, are counted up to the farthest right nonblank character. For example, if you enter 'ABb41.C', it has a length of 4 characters. If you enter 'b'4141.XYb'41.Zb'41.', it has a length of 6 characters. The parameter in this case is b'4141.XYb'41.Z.

**PDATA** This specifies whether the input from the display format is used as parameters, or is held as input for the program's first input request from the display station.

**NO** This specifies use of the input from the display as parameters for a procedure, and treatment as if the parameter had been entered with the procedure. If you do not specify PDATA, NO is assumed.

**YES** This specifies use of the input from the display format as program data that is to be given to a program on its first input request from the display station.

**UPSI** This specifies whether the current setting of the eight UPSI switches is placed in the display format indicator table.

**NO** This specifies that the switches are not used. If you do not specify UPSI, NO is assumed.

**YES** This specifies placement of the current setting of the eight UPSI switches (1 through 8) into the display format indicator table as indicators 91 through 98, respectively.

### Example 1

The following PROMPT statement displays a format named DISPLAY1 from a format load member named PROMPTS:

```
// PROMPT MEMBER-PROMPTS,FORMAT-DISPLAY1
```

### Example 2

See "Example 3: Procedure SCRNPRT" on page 2-14 for an example of a procedure using the PROMPT OCL statement.

### Example 3

The following PROMPT statements show two prompt displays:

```
// PROMPT MEMBER-SCREENS,FORMAT-DISPLAY1, PROMPT Statement for
//          LENGTH-'8,6,8,8,8'           Parameters 1-5
*
// PROMPT MEMBER-SCREENS,FORMAT-DISPLAY2, PROMPT Statement for
//          START-6,LENGTH-'',,20,'       Parameters 6-10
*
PROCA ?1?,???,?3?,?4?,?5?,?6?,?7?,?8?,?9?,?10?
```

Example Display Format Part 1 contains prompts for parameters 1 through 5. Parameter 2 has a length of 6 characters. All other parameters have lengths of 8 characters.

```

Example Display Format
      Part 1
Parameter 1 . . . . . _____
Parameter 2 . . . . . _____ (length = 6)
Parameter 3 . . . . . _____
Parameter 4 . . . . . _____
Parameter 5 . . . . . _____
    
```

Example Display Format Part 2 contains prompts for parameters 6 through 10. Parameter 9 has a length of 20 characters. All other parameters have lengths of 8 characters.

```

Example Display Format
      Part 2
Parameter 6 . . . . . _____
Parameter 7 . . . . . _____
Parameter 8 . . . . . _____
Parameter 9 . . . . . _____ (length = 20)
Parameter 10 . . . . . _____
    
```

**Example 4**

The following statements demonstrate a potential problem with assigning procedure parameters before a PROMPT OCL statement processes. In the following example, the PROMPT statement shows a prompt display that contains two fields:

- Field 1 is defined in the SFGR D-specification of the display format as an input field.
- Field 2 is defined as an input/output field. Field 2 is also defined in the SFGR D-specification to have data supplied by a program.

## RESERVE

```
// EVALUATE P1=A
// PROMPT MEMBER-SCREENS,FORMAT-DISPLAY3,LENGTH-'1,2'
*
PROCB ?1?,???
```

The following Example Display Format shows how the display appears after the PROMPT statement processes:

Example Display Format

Field 1 . . . . .

Field 2 . . . . . A

The character 'A' is in *Field 2* even though the EVALUATE PCE statement processed before the PROMPT statement assigned that value to the first parameter (P1). For the field and parameter to correspond, the operator must enter 'A' in *Field 1*. If the operator also entered the characters BC in *Field 2*, the following values would be substituted in PROCB:

```
PROCB A,BC
```

Parameters assigned before PROMPT statements must correspond to the order of the output or input/output fields defined in a display format to have data supplied by a program.

---

## REGION OCL Statement

This statement is supported for System/36 compatibility only. Syntax checking is done, but no other action is taken.

---

## RESERVE OCL Statement

This statement is supported for System/36 compatibility only. Syntax checking is done, but no other action is taken.

---

## RUN OCL Statement

The RUN OCL statement indicates the end of the OCL statements for a job step. After the system reads the RUN statement, it runs the program named in the LOAD statement.

**Placement:** Each program run on the system needs a RUN OCL statement. It is the last statement within the set of OCL statements for each job step.

```
// RUN
```

59020324-0

The RUN OCL statement has no parameters.

### Example

The following example loads and runs a program called PASTDUE:

```
// LOAD PASTDUE
// RUN
```

---

## RUN400 OCL Statement

The RUN400 OCL statement runs the specified OS/400 CL command, using the specified parameter values. OS/400 errors that occur during the running of the CL command can be evaluated using the ?MSGID? (Message ID) Procedure Control Substitution Expression.

**Placement:** Place the RUN400 OCL statement anywhere among the OCL statements except between the LOAD and RUN OCL statements.

```
// RUN400 CMD-'OS/400 CL command string'
```

R15F003.0

**CMD** This specifies the OS/400 CL command that is run, along with the appropriate parameter values. The entire expression, consisting of the command name and all parameters, must be enclosed in apostrophes ('). There is a 512 byte limit on the size of the OCL statement.

### Example

This example copies all of the members of file QS36SRC, located in the current session library, to a member of the database physical file SRCMBRS, located in library BACKUP. The members are copied in record-mode LIBRFILE format to file SRCMBRS.

```
// RUN400 CMD-'SAVS36LIBM FROMMBR(*ALL) FROMLIB(?SLIB?) +
SRCMBRS(*SRC) DEV(*PHYFILE) +
PHYFILE(BACKUP/SRCMBRS)'
```

---

## SESSION OCL Statement

The SESSION OCL statement does the following:

- Identifies the communications configuration with which the program is to communicate. The Location parameter identifies the remote location associated with the session. (The remote location is specified on the device description.)
- Identifies the symbolic identifier of the session.
- Can also optionally specify one or more parameters that change the attributes of the configuration for that session only.

**Placement:** Place the SESSION OCL statement between the LOAD and RUN OCL statements.

**Mapping:** The SESSION OCL statement uses the Override ICF Program Device Entry (OVRICFDEVE) command. See the Communications chapter of the *System/36 Environment Programming* book for a complete explanation of this mapping and for more information on the use of the SESSION OCL statement.

The SESSION OCL statement is used for the following AS/400 communications functions:

- Advanced program-to-program communications (APPC)
- SNA upline facility (SNUF)
- BSC equivalence link communications (BSCCEL)
- RPG II BSC telecommunications
- Asynchronous communications
- Finance communications
- Retail communications
- Intrasystem communications

**Note:** Finance and retail functions on System/36 were formerly configured as a single System/36 SSP-ICF finance subsystem, where you specified N (Finance) or Y (Retail) at the *Point of Sale* prompt on the subsystem configuration display. On the AS/400 system, they are divided into two separate functions: finance communications and retail communications.

The SESSION OCL statement is supported for System/36 compatibility only for the following System/36 SSP-ICF subsystems:

- System/36 SSP-ICF BSC CCP
- System/36 SSP-ICF BSC CICS
- System/36 SSP-ICF BSC IMS
- System/36 SSP-ICF PEER

These SSP-ICF subsystems are not supported on the AS/400 system. Syntax checking is done, but no other action is taken.

Only the statements for the communications types supported on the AS/400 system are shown here. See the following books for information about the support provided by the various communications types:

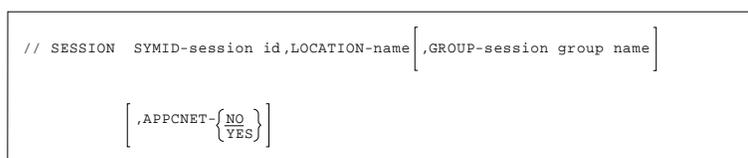
- *System/36 Environment Programming*
- *ICF Programming*
- *APPC Programming*
- *SNA Upline Facility Programming*

- *BSC Equivalence Link Programming*
- *Asynchronous Communications Programming*
- *Intrasystem Communications Programming*
- *Retail Communications Programming*
- *Finance Communications Programming*

See the *System/36-Compatible RPG II User's Guide and Reference* book for more information on RPG II telecommunications.

## For APPC and APPN

The following is the syntax diagram for APPC and APPN:



This syntax diagram has the following parameters:

### LOCATION

This specifies the remote location name to be associated with this session. The remote location name refers to the remote location with which your program is to communicate, and corresponds to the value specified for the remote location name (RMTLOCNAME) parameter on the Create Device Description APPC (CRTDEVAPPC) command during APPC configuration or the remote location name specified for the APPN remote location entry (APPNRMTE) parameter on the Create Configuration List (CRTCFGL) command during APPN configuration. This parameter is required and has no default.

### SYMID

This specifies the symbolic identifier of the session with which this SESSION statement is associated. Your program uses this identifier when it acquires the session and whenever it issues any operation in the session. The identifier must be 2 characters; the first character must be numeric (0 through 9) and the second character must be alphabetic (A through Z, \$, #, or @). This parameter is required and has no default.

### GROUP

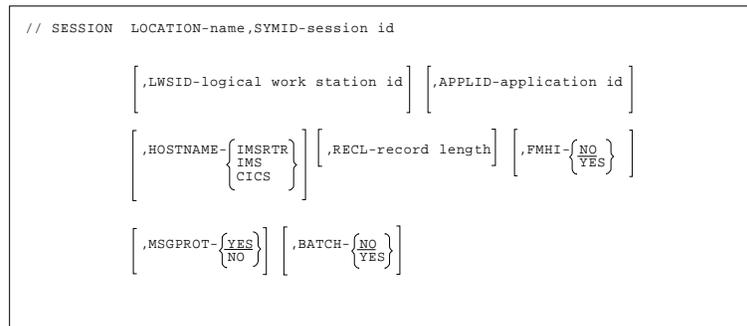
This specifies the mode to be used with this session. Mode is equivalent to System/36 session group. If you do not specify the GROUP parameter on this SESSION statement, the default mode, specified in the system's network attributes, is used.

### APPCNET

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. The system will always assume the equivalent of APPCNET-YES.

## For SNA Upline Facility

The following is the syntax diagram for SNA upline facility:



59020331-0

This syntax diagram has the following parameters:

### LOCATION

This specifies the remote location name to be associated with this session. The remote location name refers to the remote location with which your program is to communicate, and corresponds to the value specified for the remote location name (RMTLOCNAME) parameter of the Create Device Description SNUF (CRTDEVSNUF) command. This parameter is required and has no default.

### SYMID

This specifies the symbolic identifier of the session with which this SESSION statement is associated. Your program uses this identifier when it acquires the session and whenever it issues any operation in the session. The identifier must be 2 characters; the first character must be numeric (0 through 9) and the second character must be alphabetic (A through Z, \$, #, or @). This parameter is required and has no default.

### LWSID

This specifies the identifier of the logical unit for which this session is intended. The identifier must be a decimal value from 1 through 255. This value corresponds to the value specified on the local location address (LOCADR) parameter of the CRTDEVSNUF command and will therefore direct your session to a particular device description for the remote location specified. If the identifier is not specified, any available device description associated with the specified remote location name, except a device description that is reserved for program start requests, can be used. A device description can be reserved for program start requests by specifying \*YES for the program start request capable (PGMSTRRQS) parameter on the CRTDEVSNUF command.

### APPLID

This specifies the VTAM\* identifier of the CICS/VS or IMS/VS host subsystem to be sent with the log-on message. If this parameter is not specified, the value specified on the application identifier (APPID) parameter of the CRTDEVSNUF command will be used.

### HOSTNAME

This specifies the type of host subsystem with which this session is to communicate. If this parameter is not specified, the host system type specified on the host type (HOST) parameter of the CRTDEVSNUF command will be used.

- CICS** This indicates that the session is to communicate with CICS/VS.
- IMS** This indicates that the session is to communicate with IMS/VS.
- IMSRTR** This indicates that the session is to communicate with IMS/VS using the ready-to-receive option. See the *SNA Upline Facility Programming* book for more information on the ready-to-receive option.
- RECL** This specifies the maximum record length that will be sent or received for this session. The record length can be any decimal value from 1 through 4075 bytes. If this parameter is not specified, the value specified on the record length (RCDLEN) parameter of the CRTDEVSNUF command will be used.
- FMHI** This specifies, for both CICS/VS and IMS/VS, whether received function management headers should be passed to the program. The default is NO.
- YES** This indicates that function management headers should be passed.
- NO** This indicates that function management headers will be removed before the data is passed to the program.
- MSGPROT** This specifies, for both CICS/VS and IMS/VS, whether message protection is to be used for this session. YES is the default and is only valid when BATCH-NO is also specified.
- YES** This indicates that message protection will be used and that messages will be saved until they are acknowledged. If an error occurs, SNUF will attempt to resynchronize with the remote system. For more information on session recovery, refer to the *SNA Upline Facility Programming* book.
- NO** This indicates that message protection will not be used.
- BATCH** This specifies, for both CICS/VS and IMS/VS, whether this session will be used for batch activity. See the *SNA Upline Facility Programming* book for information on batch processing. The default is NO.
- YES** This indicates that batch activity will occur.
- NO** This indicates that no batch activity will occur.

## For BSCEL and RPG II Telecommunications

The SESSION OCL statement is used for applications using the BSCEL support. Since RPG II programs using the telecommunications specification implicitly use the ICF BSCEL support, you must use a SESSION OCL statement in the job stream. This replaces the COMM OCL statement used by the System/36, System/34, and System/32.

```

// SESSION LOCATION-name,SYMID-session id [ ,PARTNER- {ATTR
                                           {NORM} }

[ ,SWTYP- {MC
           {AA
           {MA} } ] [ ,PHONE-member name ] [ ,REFRESH- {YES
                                                         {NO} } ]

[ ,RESTORE- {NO
             {YES} } ] [ ,RECL-record length ] [ ,BLKL-block length ]

[ ,RECSEP-separator character ] [ ,ITB- {YES
                                          {NO} } ] [ ,BLANK- {C
                                                             {T
                                                             {N} } ]

[ ,TRANSP- {YES
            {NO} } ] [ ,LIBRARY-name ]

```

S9020326-0

Figure 5-1. SESSION Syntax for BSCEL

When any of the following parameters, except LOCATION, SYMID, PHONE, REFRESH, RESTORE, and LIBRARY, are specified on the SESSION statement, the parameter value overrides the corresponding parameter specified during configuration. The configuration values are not overridden if the corresponding parameters are not specified on the SESSION statement.

#### LOCATION

This specifies the remote location name to be associated with this session. The remote location name refers to the remote location with which your program is to communicate, and corresponds to the value specified for the remote location name (RMTLOCNAME) parameter of the Create Device Description BSC (CRTDEVBS) command. This parameter is required and has no default.

#### SYMID

This specifies the symbolic identifier of the session with which this SESSION statement is associated. Your program uses this identifier when it acquires the session and whenever it issues any operation in the session. The identifier must be 2 characters; the first character must be numeric (0 through 9) and the second character must be alphabetic (A through Z, \$, #, or @). This parameter is required and has no default. If your RPG II application is using telecommunications specifications, you must specify 9# for this parameter.

#### PARTNER

This specifies the partner attribute of this session. If the PARTNER parameter is not specified, the value specified on the remote BSCEL (RMTBSCEL) parameter of the CRTDEVBS command will be used. If your RPG II application is using telecommunications specifications, specify ATTR for the PARTNER parameter.

**NORM** This indicates that BSCEL messages and commands can be recognized by the remote system. This system is usually a System/34, System/36, or another AS/400 system using BSCEL.

**ATTR** This indicates that the BSC/CEL commands and messages cannot be handled by the remote system. This remote system is usually a device such as a 3741 Data Entry Station.

### Line Connection Parameters

The SESSION statement for BSC/CEL has the following line connection parameters:

**SWTYP** This specifies the method of making a connection on the line for the session being established. This parameter only applies when a switched line is being used.

If the SWTYP parameter is not specified on the SESSION statement, the switch type from the initial connection (INLCNN) parameter of the Create Controller Description BSC (CRTCTLBSC) command will be used.

The values that can be specified are:

**MC** This indicates that the local system starts the call. To get Manual Call, you must also specify the autodial (AUTODIAL) parameter as \*NO on the Create Line Description BSC (CRTLINBSC) or Change Line Description BSC (CHGLINBSC) command.

**AA** This indicates that the remote system starts the call, and the local system answers the call. To get Auto Answer, you must also specify \*YES for the autoanswer (AUTOANS) parameter on the CRTLINBSC or CHGLINBSC command.

**MA** This indicates that the remote system starts the call, and the local system answers the call. To get Manual Answer, you must also specify \*NO for the AUTOANS parameter on the CRTLINBSC or CHGLINBSC command.

**PHONE** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. The AS/400 auto dial function does not support a phone list. However, a single number can be used by specifying \*YES on the AUTODIAL parameter of the CRTLINBSC command and the requested phone number on the connection number (CANNBR) parameter of the CRTCTLBSC command.

### REFRESH

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. The AS/400 automatic-dial function should be used.

### RESTORE

This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. The AS/400 automatic-dial function should be used.

**LIBRARY** This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. The AS/400 automatic-dial function should be used.

## Data Record Parameters

The SESSION statement for BSCCL has the following data record parameters:

**RECL** Specifies the length of the longest record you expect to send or receive in this session. You can enter a length of from 1 through 4075 bytes. If PARTNER-NORM is specified, some BSCCL messages may be truncated if the RECL parameter value is less than 90 bytes. If the RECL parameter is not specified, the value specified on the record length (RCDLEN) parameter on the CRTDEVBS command will be used.

**BLKL** This specifies the length, from 1 through 4075, of the block of data records you expect to send or receive in this session. The block length must be at least as long as one record. If you specify 0, data records are not blocked. If the BLKL parameter is not specified, the value specified on the block length (BLKLEN) parameter on the CRTDEVBS command will be used.

**RECSEP** This specifies the record separator character, in hexadecimal, used to separate data records when the records are blocked. You can enter any ASCII or EBCDIC hexadecimal character except the data link control characters, which are used to control communications over a BSC line. Refer to the *BSC Equivalence Link Programming* book for more information on the data link control characters.

If you specify 00, no record separator is used.

If the RECSEP parameter is not specified, the value specified on the separator character (SEPCHAR) parameter on the CRTDEVBS command will be used.

**ITB** This specifies whether intermediate text block (ITB) characters are to be inserted between data records when the data is blocked. The ITB character causes error checking after each record in a block and at the end of each block.

If the ITB parameter is not specified, the value specified on the blocking type (BLOCK) parameter on the CRTDEVBS command will be used.

**YES** This indicates error checking is to be performed after each record in a block has been received.

**NO** This indicates error checking is to not be performed after each record in a block has been received.

**BLANK** This specifies blank truncation or blank compression.

If the BLANK parameter is not specified, the value specified on the compress and decompress data (DTACPR) and truncate trailing blanks (TRUNC) parameters on the CRTDEVBS command will be used.

**T** This indicates that trailing blanks are to be truncated from each data record.

**C** This indicates that each series of three or more blanks in each record is to be compressed. The blanks are replaced when the data is received.

**N** This indicates that neither blank truncation nor blank compression is to be done.

**TRANSP** This specifies whether data will be sent in transparent text mode.

If the TRANSP parameter is not specified, the value specified on the transmit in transparent mode (TRNSPY) parameter on the CRTDEVBS command will be used.

**YES** This indicates that data will be sent in transparent mode.

If you are using ASCII, blank compression, or record separators, your program will receive an 82xx return code when it attempts an acquire operation.

**NO** This indicates that data will not be sent in transparent mode.

## For Asynchronous Communications

The following is the syntax diagram for asynchronous communications:

```
// SESSION LOCATION-name,SYMID-session id
```

59020333-0

This syntax diagram has the following parameters:

### LOCATION

This specifies the remote location name to be associated with this session. The remote location name refers to the remote location with which your program is to communicate, and corresponds to the value specified for the remote location name (RMTLOCNAME) parameter of the Create Device Description Asynchronous (CRTDEVASC) command. This parameter is required and has no default.

### SYMID

This specifies the symbolic identifier of the session with which this SESSION statement is associated. Your program uses this identifier when it acquires the session and whenever it issues any operation in the session. The identifier must be 2 characters; the first character must be numeric (0 through 9) and the second character must be alphabetic (A through Z, \$, #, or @). This parameter is required and has no default.

## For Finance Communications

The following is the syntax diagram for finance communications:

```
// SESSION LOCATION-name,SYMID-session id,LWSID-logical work station ID
```

59020683-0

This syntax diagram has the following parameters:

### LOCATION

This specifies the remote location name to be associated with this session. The remote location name refers to the remote location with which your program is to communicate, and corresponds to the value specified for the remote location name (RMTLOCNAME) parameter of the Create Device Description Finance (CRTDEVFNC) command.

## SESSION

The AS/400 Finance communications support requires that each remote location name be unique. If you use multiple logical work station IDs on one location, or multiple subsystems that have the same remote location name, you must develop a remote location naming scheme to ensure each name is unique, and then edit your finance communications SESSION OCL statements to conform to the new naming scheme. This parameter is required and has no default.

- SYMID** This specifies the symbolic identifier of the session with which this SESSION statement is associated. Your program uses this identifier when it acquires the session and whenever it issues any operation in the session. The identifier must be 2 characters; the first character must be numeric (0 through 9) and the second character must be alphabetic (A through Z, \$, #, or @). This parameter is required and has no default.
- LWSID** This specifies the identifier of the logical work station ID (logical unit) for which this session is intended. The identifier must be a decimal value between 1 and 255. This value corresponds to the value specified on the local location address (LOCADR) parameter of the CRTDEVFNC command and directs your session to a particular device description for the remote location specified. Since the AS/400 Finance communications support requires unique remote location names, this parameter is not required.

## For Retail Communications

The following is the syntax diagram for retail communications:

```
// SESSION LOCATION-name,SYMID-session ID,LWSID-logical workstation ID  
[.MAXMSG-NNNN][,APPLID-application id][,BATCH-  
{NO  
YES}]
```

59020684-0

This syntax diagram has the following parameters:

### LOCATION

This specifies the remote location name to be associated with this session. The remote location name refers to the remote location with which your program is to communicate, and corresponds to the value specified for the remote location name (RMTLOCNAME) parameter of the Create Device Description Retail (CRTDEVRTL) command.

The AS/400 Retail communications support requires that each remote location name be unique. If you use multiple logical work station IDs on one location, or multiple subsystems that have the same remote location name, you must develop a remote location naming scheme to ensure each name is unique, and then edit your retail communications SESSION OCL statements to conform to the new naming scheme. This parameter is required and has no default.

- SYMID** This specifies the symbolic identifier of the session with which this SESSION statement is associated. Your program uses this identifier when it acquires the session and whenever it issues any operation in the session. The identifier must be 2 characters; the first character must be numeric (0 through 9) and the second character must be alpha-

betic (A through Z, \$, #, or @). This parameter is required and has no default.

- LWSID** This specifies the identifier of the logical work station ID (logical unit) for which this session is intended. The identifier must be a decimal value between 1 and 254. This value corresponds to the value specified on the local location address (LOCADR) parameter of the CRTDEVRTL command and directs your session to a particular device description for the remote location specified. Since the AS/400 Retail communications support requires unique remote location names, this parameter is not required.
- MAXMSG** This parameter is ignored. On System/36, this specified the SNA session pacing values for both inbound and outbound data. You must specify pacing on the pacing parameter of the CRTDEVRTL command or the Change Device Description Retail (CHGDEVRTL) command to use SNA pacing.
- APPLID** This parameter specifies the name of the application to be attached to or to be started on the point-of-sale retail controller. This parameter is optional.
- BATCH** This specifies whether the point-of-sale application has a communications session with your program. This parameter is optional.
- NO** The default is NO. The point-of-sale application is attached to your program and a communications session is established when your program sends an acquire operation.
- YES** The retail communications support attempts to start the application program specified in the APPLID parameter of the SESSION OCL statement on the point-of-sale retail controller when your program sends an acquire operation. No communications session is established between your program and the point-of-sale application on the controller.

## For Intrasystem Communications

The following is the syntax diagram for intrasystem communications:

```
// SESSION LOCATION-name,SYMID-session id

[ ,BATCH- { NO }
           { YES } ]
```

59020706-0

This syntax diagram has the following parameters:

### LOCATION

This specifies the remote location name to be associated with this session. The remote location name refers to the remote location with which your program is to communicate and corresponds to the value specified for the remote location name (RMTLOCNAME) parameter of the Create Device Description Intrasystem (CRTDEVINTR) command. This parameter is required and has no default.

- SYMID** This specifies the symbolic identifier of the session with which this SESSION statement is associated. Your program uses this identifier when it acquires the session and whenever it issues any operation in the session. The identifier must be 2 characters; the first character must be numeric (0 through 9), and the second character must be alphabetic (A through Z, \$, #, or @). This parameter is required and has no default.
- BATCH** This specifies whether batch-oriented operations (negative-response, cancel, and function-management-header functions) can be issued for this session.
- YES** Indicates they can be issued.
- NO** Indicates they cannot be issued. NO is the default.

---

## SETDEV OCL Statement

The SETDEV OCL statement allows you to change the AS/400 device names associated with OCL device names. For example, you can change the I1, T1, T2, and TC names during a System/36 environment session or job. When you do this, you do not impact other System/36 environment sessions or jobs. This device override utilizes any diskette or tape device on the AS/400 system, while using System/36 environment OCL statements and functions. If you want to change device names without the SETDEV OCL statement, you have to end all System/36 environment sessions, and make permanent changes to the configuration.

**Placement:** This OCL statement may be specified anywhere OCL statements are allowed.

```
// SETDEV UNIT-[ I1
                 T1
                 T2
                 TC
                 'T1,T2'
                 etc. ] ,DEV0-[ DEV0 name
                               0
                               'DEV01,DEV02'
                               etc. ] [ ,SESSION- { NO
                                                  YES }
```

RV2P184-2

**UNIT** This specifies the System/36 environment device ID to be re-assigned. You may specify a single device or a list of devices. If you specify a list of devices, the list must start and end with a single quote with commas delimiting the list elements. The maximum number of elements in the list is four and each element must be unique (no duplicates). This is a required parameter.

**DEV0** This specifies the AS/400 device description name to associate with the UNIT parameter. You may specify a single device description name or a list of device description names. If you specify a list of device descriptions, the list must start and end with single quotes and must be delimited with a comma. If a list is specified, the number of devices in the list must be the same as the number of devices in the UNIT parameter. Devices will be assigned, in order, to the corresponding UNIT value. A zero (0) indicates this device is not configured.

**Note:** Only diskette device description names will be allowed for UNIT-I1 and only tape device descriptions will be allowed for

UNIT-T1, UNIT-T2, or UNIT-TC. No duplicates are allowed in the list with the exception of the list value zero (0).

**SESSION** This specifies the scope of the SETDEV statement. A value of YES indicates the values specified on the statement will persist for the rest of the session or until another SETDEV statement with a SESSION-YES value specified is processed. If SESSION-YES is specified in a procedure, the statement will not take effect until the end of the outermost procedure. A value of NO indicates that the values in the SETDEV statement persist until the job ends or until another SETDEV statement with SETDEV-NO is specified. If the SESSION parameter is not specified, NO is assumed.

---

## START OCL Statement

The START OCL statement causes the specific spooling writer to start. If you have \*JOBCTL authority, you can use the START OCL statement to do the following:

- Start printing spool file entries for a specific printer or for all printers.
- Start printing spool file entries with a specific forms number for a specific printer or for all printers.
- Start printing spool file entries for a specific printer or for all printers such that those entries using the same forms are printed together.

**Placement:** Place the START OCL statement anywhere among the OCL statements. You can evoke or submit the START OCL statement to the job queue.

```
// START PRT, [ ALL
(P) [ printer id
[ system printer ] ], [ FORMS
[ forms number ] ]
```

59020617-0

**PRT or P** This starts a spool writer for a printer. This means that when the printer is available, spool file entries can print. You can use the PRT parameter to start the spooling writer after you enter a STOP PRT OCL statement or a STOP PRT control command.

If you do not enter a second parameter, the system printer is assumed. If you do not enter a third parameter, the spooling writer prints the available entries according to their position in the spool file.

**ALL** This specifies that the spooling writers for all printers are started.

**printer id** This specifies the 2-character ID of the printer for which the spool writer starts. If you do not specify a parameter, the system printer is assumed. You must have \*JOBCTL authority.

**FORMS** This specifies that the spooling writer print all available spool file entries that require the forms currently used in the printer, regardless of the entry's position in the spool file. After all such entries print, the operator controlling the printer is prompted to change the forms, and all entries using the next group of forms print. Printing all entries reduces the number of operator messages to change the forms.

# STOP

## forms number

This specifies the 1- to 4-character forms number used by the spooling writer. Only the spooling file entries using the specified forms number print.

## Example 1

The following example starts the spooling writer for printer P1 in order to start printing all spool file entries with forms number 1324 (that is, they are to be printed on printout form 1324):

```
// START PRT,P1,1324
```

or:

```
// START P,P1,1324
```

## Example 2

The following example starts the spool writer for the system printer. All spool file entries that require the forms currently in use by the system printer, print.

```
// START P,,FORMS
```

After all such entries print, the operator controlling the printer is prompted to change the forms, and the next group of spool file entries print.

---

## STOP OCL Statement

If you have \*JOBCTL authority and control one or more printers, you can use the STOP OCL statement to stop the printing of all entries from the spool file for a specific printer, or for all printers. You can stop printing immediately, at the end of the currently printing page, or at the end of the currently printing spool file entry.

**Placement:** Place the STOP OCL statement anywhere among the OCL statements. You can evoke or submit the STOP OCL statement to the job queue.

```
// STOP PRT, [ ALL ] , [ PAGE ]  
 (P) [ printer id ] , [ JOB ]  
 [ system printer ]
```

59020618-0

**PRT or P** This specifies that printing spool file entries stops. If a spool file entry prints, and you do not specify a third parameter, printing stops immediately. Resume printing by typing the START OCL statement or the START PRT control command.

**ALL** This specifies that printing stops for all printers.

**printer id** This specifies the 2-character ID of the printer for which the spool writer stops. If you do not specify a printer ID, the system printer is assumed.

**PAGE** This stops the spooling writer when it has completed printing the current page.

**JOB** This stops the spooling writer when it completes printing the current copy of the spool file entry.

**Example**

The following examples stop the spooling writers for all printers that you control when the spool file entry currently printing on each of them completes:

```
// STOP PRT,ALL,JOB
```

or:

```
// STOP P,ALL,JOB
```

---

**SWITCH OCL Statement**

The SWITCH OCL statement sets one or more of the user program status indicator (UPSI) switches for the display station to on (1, one) or off (0, zero). The switch setting remains in effect until one of the following conditions exist:

- Another SWITCH statement processes.
- The display station session ends.
- A program changes the setting of any of the indicators.

All switches are set to off when a display station session begins (when an operator signs on).

A job placed on the job queue uses a copy of the switches as they existed when the job was placed on the queue. A job that was EVOKED or a job RELEASED using ATTR OCL also uses a copy of the switches for the display station as they existed when the job was evoked or released. To determine the switch settings from a procedure, see “SWITCH OCL Statement.”

If an SSP procedure changes the setting of a switch, the switch returns to its original setting when the SSP procedure ends. A set of switches also exists for each MRT procedure that is running.

**Placement:** Place the SWITCH OCL statement anywhere among the OCL statements.

```
// SWITCH  switch settings
```

59020335-0

**switch settings**

This specifies how to set the switches, and consists of 8 characters, one for each of the eight switches. The first, or farthest left, character gives the setting of switch 1, the second character gives the setting of switch 2, and so on.

The parameter must always contain 8 characters. For each indicator, you must use one of the characters shown in the following table:

Character	Meaning
0 (zero)	Set the indicator off
1 (one)	Set the indicator on
X	Leave the indicator as it is

**Example**

The following example causes the results listed in the table in Table 5-1:

```
// SWITCH 1X0110XX
```

Table 5-1. Example Switch Settings

Indicator	Switch Setting	Result
1	1	Set on
2	X	Unaffected
3	0	Set off
4	1	Set on
5	1	Set on
6	0	Set off
7	X	Unaffected
8	X	Unaffected

**SYSLIST OCL Statement**

The SYSLIST OCL statement changes the printing or displaying of system list output. System list output is the output created by all System/36 environment utility programs except for the data communications utility programs.

The SYSLIST statement causes the system list output to be:

- Listed on the printer that was assigned to the display station during system configuration
- Listed on one of the other printers
- Displayed at the display station
- Not listed at all

The SYSLIST statement remains in effect until the display station session ends or until it is changed by:

- Another SYSLIST OCL statement
- The SYSLIST procedure

A PRINTER OCL statement with NAME-\$SYSLIST specified further controls the system list output for one job step. In this case, the device specified on the PRINTER statement is used, not the device specified on the SYSLIST statement. See "PRINTER OCL Statement" on page 5-68 for more information.

**Notes:**

1. If the operator runs the SYSLIST OCL statement during inquiry mode by pressing the System Request key, the change made by the SYSLIST OCL statement is in effect only during inquiry mode.
2. The SYSLIST OCL statement does not change the system list device if debugging is on. For more information, see "DEBUG OCL Statement" on page 5-22.

**Placement:** You can place the SYSLIST statement anywhere among the OCL statements.

```
// SYSLIST { CRT
            { PRINTER
              { printer id
                OFF
              }
            }
          } { EXTN
            { NOEXTN
              }
          } { FOLD
            { NOFOLD
              }
          }
```

59020336-0

- CRT** This specifies that the system list output is displayed at the requesting display station.
- After seeing each system list display, the operator can request to see the next set of lines, or that the display end. If the operator enters a space, the number of lines shown on the next display is the same as the number of lines shown on the previous display. If the operator enters 0, the display ends. If the operator enters any characters other than a space or 0 through 18, the next set of lines appears.
- System output from MRT programs prints on the system printer. System list output from programs that release the requesting display station, and from programs that are run from the job queue, prints on the session printer.
- PRINTER** This specifies printing system list output on the printer that was assigned to the display station during system configuration or on the printer assigned by the SET procedure or \$SETCF utility program.
- printer id** This specifies printing system list output on the printer with the specified printer ID. Use the STATUS WORKSTN command to determine the printer IDs. To print output on the system printer, specify the printer ID of the system printer.
- OFF** This specifies that no system list output appears or prints.
- EXTN** This specifies printing or displaying the extended characters in the system list output.
- NOEXTN** This specifies that the extended characters not print or display in the system output. The system-defined double-byte character is listed for any extended characters.
- Note:** Use the EXTN and NOEXTN parameters for the DBCS version of the OS/400 program. The OS/400 program ignores them for non-DBCS systems. When a session begins, extended character processing for system list output is assumed. During the session, the operator can use the EXTN and NOEXTN parameters to turn on and turn off extended character processing for system list output.
- FOLD** This specifies that system list output displayed at the display station is not truncated if it exceeds the column width of the display station. Instead, data continues on the next line of the display.
- NOFOLD** This specifies truncation of system list output displayed at the display station if it exceeds the column width of the display station. If you do not specify a parameter, NOFOLD is assumed.

**Example**

The following example assigns printer P2 to list system list output:

```
// SYSLIST P2
```

**TIMERSET OCL Statement**

The TIMERSET OCL statement allows a user with sufficient authority to indicate whether a system should power on and perform an unattended initial program load (IPL). Three types of unattended IPL operations are supported by the OCL statement:

- An unattended IPL after a power failure
- An unattended IPL from a remote site using a communications modem
- An unattended IPL at a specific time and date

To use the TIMERSET OCL statement to either enable or disable any of the three unattended IPL operations, a user must have both \*ALLOBJ and \*SECADM authority, as well as \*USE authority to the CHGSYSVAL CL command. In order for an unattended IPL to occur, the system must have either lost and subsequently regained power or presently be turned off. The keylock switch on the control panel must be set to either the **Normal** or the **Auto** position. Following an unattended IPL, the Sign On display appears.

The TIMERSET OCL statement also allows any user to determine whether the system has been previously prepared to perform any of the three unattended IPL operations.

**Placement:** The TIMERSET OCL statement can appear anywhere among the OCL statements.

```
// TIMERSET [ QUERY- YES
              TIMED- { YES }
                   { NO }
              RESTART- { YES }
                   { NO }
              REMOTE- { YES }
                   { NO } ] [ ,DATE- { mddy }
                              { dmyy }
                              { yymmdd } ] [ ,TIME- hhmmss ]
```

RV2P179-0

**QUERY-YES**

Specifies that the current system values for the three unattended IPL operations should be listed on the current system list device.

**TIMED**

Specifies whether the system should power on and perform an unattended IPL at a specific time and date.

**YES**

Specifies that the system should power on and perform an unattended IPL at a specific time and date. The DATE and TIME parameters are required when TIMED-YES is specified.

**Note:** The system value set when the TIMED-YES, DATE and TIME parameters and values are used is not cleared when the specified date and time are reached and the unattended IPL takes place. The system value remains as set until it is changed through another use of the TIMED-YES, DATE and TIME parameters and values.

**NO**

Specifies that the system should not power on and perform an unattended IPL at a specific time and date.

**RESTART**

Specifies whether the system should power on and perform an unattended IPL after a power failure occurs.

**YES**

Specifies that the system should power on and perform an unattended IPL after a power failure occurs.

**NO**

Specifies that the system should not power on and perform an unattended IPL after a power failure occurs.

**REMOTE**

Specifies whether the system should power on and perform an unattended IPL when a telephone call is received through a modem.

**YES**

Specifies that the system should power on and perform an unattended IPL when a telephone call is received through a modem.

**NO**

Specifies that the system should not power on and perform an unattended IPL when a telephone call is received through a modem.

**DATE**

Specifies the month, day, and year that the system should power on and perform an unattended IPL. If the date specified is a past date, an error message is sent. This parameter can only be specified when TIMED-YES is also specified. The DATE parameter value is specified as follows:

**mmddy, ddmmy, yymmdd**

Specifies the month (mm), day (dd), and year (yy) that the system should power on and perform an unattended IPL.

**Note:** The month, day, and year must be specified in the session date format.

**TIME**

Specifies the time of day in hours, minutes and seconds (24-hour format) that the system should power on and perform an unattended IPL. (The value specified for seconds is ignored.) The time must be at least five minutes greater than the current time; otherwise, an error message is sent. This parameter can only be specified when TIMED-YES is also specified. The TIME parameter value is specified as follows:

**hhmmss**

Specifies the hour (hh), minute (mm), and second (ss) of the specified date that the system should power on and perform an unattended IPL.

### Example

This example shows how to prepare a system to perform an unattended power on and IPL at a specific time and date.

```
// TIMERSET TIMED=YES,DATE-121291,TIME-063000
```

---

## VARY OCL Statement

The VARY OCL statement allows you to change the status of the following from online to offline and from offline to online:

- Display stations
- Printers
- The system printer
- The diskette drive
- Tape drives

Operators or programs cannot use devices that are offline. You cannot use the VARY OCL statement to take offline a device that is allocated to a program or signed on. If you have \*JOBCTL authority, you can enter the VARY OCL statement from any type of display station.

**Note:** On System/36, when a display or printer is varied on, the line and controller are also varied on. On the AS/400 system, varying a device online does not vary the associated line and controller online. To vary on a device, you must ensure that the line and controller are varied on.

**Placement:** Place the VARY OCL statement anywhere among the OCL statements except between the LOAD and the RUN OCL statements.

```
// VARY {ON,OFF} {
    display id
    printer id
    PRT
    (P)
    I1
    T1
    T2
    TC
    controller id
    ,line number
    controller id,line number
}
```

59020337-2

**ON** This specifies placing the specified device online.

**OFF** This specifies placing the specified device offline.

#### display id

This specifies the System/36 environment 2-character ID of a display station placed online or offline. Use the Display System/36 (DSPS36) command to determine the display station IDs.

You must sign a display station off before placing it offline.

**printer id** This specifies the System/36 environment 2-character ID of a printer placed online or offline. Use the Display System/36 (DSPS36) command to determine the printer IDs.

The spooling writers for a printer must be stopped or printing complete in order to place the printer offline. Use the STATUS WRT command to determine whether a spooling writer is stopped or printing complete.

- PRT or P** This specifies placing the system printer online or offline.
- I1** This specifies placing the diskette drive online or offline.
- T1** This specifies placing tape drive 1 online or offline.
- T2** This specifies placing tape drive 2 online or offline.
- TC** This specifies placing the tape cartridge drive online or offline.

**controller id**

Not supported. You can use the Work with Configuration Status (WRKCFGSTS) command or the Vary Configuration (VRYCFG) command to vary the controllers.

**line number**

Not supported. You can use the WRKCFGSTS command or the VRYCFG command to vary the lines.

**controller id, line number**

Not supported. You can use the WRKCFGSTS command or the VRYCFG command to vary the lines.

**Example**

The following example places device W2 online:

```
// VARY ON,W2
```

---

## WAIT OCL Statement

The WAIT OCL statement causes a job to wait until a specified time of day, or until a specified period of time passes. Once a WAIT OCL statement processes, the job does not resume processing until the specified condition is met. Either of the following actions ends the job:

- The operator who started the job can press the System Request key and select an option to cancel the job.
- The system operator can use the ENDJOB command to end the job.

Use caution with this statement. For example, if you place a job that contains this statement on the job queue, you can cause that job to prevent other jobs on the queue from processing.

**Placement:** Place the WAIT statement anywhere among the OCL statements. Place the WAIT statement in a procedure. You cannot enter it at the keyboard.

```
// WAIT {TIME-hhmmss
        {INTERVAL-hhmmss}}
```

59020338-0

**TIME** This specifies that the job stops and waits until the specified time of day occurs. **hhmmss** specifies the time of day when the job resumes processing. You must enter a 6-digit number.

## WAIT

The actual time when a job resumes processing depends on system work load and processing time. For example, if the WAIT OCL statement says // WAIT TIME-080000, and you submit the job at 075959, it is possible that although should start at 080000, it can wait until 080000 the next day due to processing time. Split-second accuracy cannot be guaranteed on a busy system.

**hh** This specifies the hour. The value must be from 00 to 24. The values 00 to 12 represent 12 midnight to 12 noon. The values 13 to 24 represent 1 p.m. through 12 midnight. The system considers both 000 000 and 240 000.

**mm** This specifies the minute. The value must be from 00 through 59.

**ss** This specifies the second. The value must be from 00 through 59.

You can specify a maximum value of 240 000. Therefore, if you specify 24 hours, the minutes and seconds values must be 00.

### INTERVAL

This specifies the amount of time a job waits before resuming processing.

**hhmmss** This specifies how long the job waits before it resumes processing. Enter a 6-digit number.

**hh** This specifies the hours. The value must be from 00 through 24.

**mm** This specifies the minute. The value must be from 00 through 59.

**ss** This specifies the second. The value must be from 00 through 59.

You can specify a maximum value of 240 000. Therefore, if you specify 24 hours, the minutes and seconds values must be 00.

### Example 1

If the program PROGA can run only at 9 p.m. (time 210000), the procedure to run PROGA could contain the following statements:

```
// WAIT TIME-210000
// LOAD PROGA
// RUN
```

### Example 2

The following example specifies that the procedure waits 1 minute between job steps:

```
// LOAD PROG1
// RUN
// WAIT INTERVAL-000100
// LOAD PROG2
// RUN
```

### Example 3

The following example specifies that a procedure waits until 4 p.m. If the time is already greater than 4 p.m., the WAIT statement does not process. For example, if the procedure ran at 5 p.m., the procedure PROC1 runs immediately. If the ?TIME? test were not performed, procedure PROC1 would not run until 4 p.m. of the *next day*.

```
// IFF ?TIME?>160000 WAIT TIME-160000
PROC1
```

---

## WORKSTN OCL Statement

The WORKSTN OCL statement supplies the system with information about a display station that a program uses. The system can assign a display station for use by a program in one of the following ways:

- The system automatically acquires the requesting display station, unless you place the program on the job queue or evoke it. The system does not require a WORKSTN OCL statement for the requesting display station.
- The system acquires the specified display station if you specify REQD=YES on a WORKSTN OCL statement.
- The program can perform the acquire operation. If the program acquires a display station, a WORKSTN statement is optional. Although the program acquires the display station, you can use a WORKSTN statement to specify information about the display station.

**Placement:** Place the WORKSTN statement anywhere among the OCL statements. All parameters, except the RESTORE parameter, take effect immediately and remain in effect until the end of the current job step. The RESTORE parameter takes effect immediately and remains in effect until the job ends or until the system releases the display station. If the WORKSTN statement appears after the last job step in a job, the WORKSTN statement parameters take effect.

```
// WORKSTN UNIT-display id [ ,SYMID-symbolic ws id ] [ ,REQD- $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$  ]
[ ,RESTORE- $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$  ] [ ,PRINT- $\left\{ \begin{array}{l} \text{printer id} \\ \text{NO} \end{array} \right\}$  ] [ ,BORDER- $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$  ]
[ ,HEADER- $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$  ] [ ,EXTN- $\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$  ]
```

59020339-0

**UNIT** This specifies the 2-character work station ID of the display station used by the program. Use the STATUS WORKSTN command to determine the display station IDs.

**SYMID** This specifies the symbolic work station ID that the program uses. The symbolic work station ID is a 2-character ID (the first character must not be numeric) that the program uses to refer to the display station. The symbolic ID cannot be the same as the work station ID assigned to any other display station or printer on the system. The symbolic work station ID can be the same as the work station ID specified by the UNIT parameter. If you do not specify SYMID, the symbolic work station ID is

assumed to be the same as the work station ID specified in the UNIT parameter.

**REQD** This specifies whether the System/36 environment or the application program acquires the display station (if a display station needs to be acquired). If the display station is the requesting display station, it does not need to be acquired.

**NO** This specifies that the application program acquires the display station. If you do not specify the REQD parameter, REQD-NO is assumed.

**YES** This specifies that the System/36 environment acquires the display station.

**RESTORE**

This specifies whether to restore the command display (or the menu display if a menu is in use) when the program releases a command display station or when the job ends.

**YES** This specifies replacement of the last information the program displays by the command or menu display on a command display station, or by the standby display on a data display station.

**NO** This specifies that the last information the program displays remains on the display screen. The operator must press the Enter key to restore the command or menu.

If you do not specify RESTORE and the last display station operation was an input to the program, the command or menu is restored. If you do not specify RESTORE and the last display station operation was an output from the program, the command or menu is not restored.

**PRINT** This defines the Print key destination for this work station. This is supported for System/36 compatibility only. Syntax checking is done, but the value is not used. Print key output is always sent to the printer defined as the session printer.

**BORDER** This determines whether a border prints around the display image when the Print key is pressed.

**YES** This specifies that a border is to be printed.

**NO** This specifies that a border is not to be printed.

**HEADER** This parameter determines whether a heading prints at the top of the page containing the display image when the Print key is pressed.

**YES** This specifies that a heading is to be printed.

**NO** This specifies that a heading is not to be printed.

**EXTN** This specifies whether to call the extended character task for processing extended characters in the data stream. The EXTN parameter is for the DBCS version of the OS/400 operating system and is ignored for non-DBCS systems.

**ON** This specifies calling the extended character task to process any extended characters in the data stream that appears.

**OFF** This specifies that the extended character task is not called. The system-defined default double-byte character appears for extended characters in the data stream. If you do not specify the EXTN parameter, ON is assumed.

**Note:** The EXTN parameter does not affect characters you enter from the keyboard at the display station.

**Example**

In the following example, display station W3 is acquired for use by the program PROGA. The program uses the symbolic name A1. The command, menu, or standby display restores when the display station releases. If the operator presses the Print key, displayed information prints on printer P2.

```
// LOAD PROGA
// WORKSTN UNIT=W3,SYMID=A1,REQD=NO,RESTORE=YES,PRINT=P2
// RUN
```

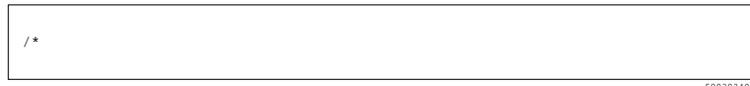
---

**/\* (End of Data) Statement**

The /\* statement indicates the end of data entered from the keyboard, or the end of in-line source data.

**Placement:** You must place a /\* statement as the last line of data you enter from the keyboard.

```
/*
```



The /\* statement has no parameters. Enter it in columns 1 and 2.

**Example**

The following example shows an end-of-data statement:

```
/*
```

**/\* (End of Data)**

## Chapter 6. Control Commands

This chapter describes the System/36 control commands that are supplied as part of the System/36 environment that the AS/400 system supports. These commands control the operation and display the status of the system, printers, and display stations. You can enter control commands from the keyboard or select them with a menu option. You cannot code them as part of a procedure. The control commands can be run from any workstation; they are not restricted to the system console.

Because the AS/400 system is different from the System/36, the function provided by some of the control commands is not needed or the function is provided by an AS/400 command. When one of these control commands is used, a message is sent to the workstation operator. For those commands where the function is not needed, the message states that the command is not supported. For those control commands where the function is provided by an AS/400 command, the message identifies the command and a reply option of 0 is allowed. For those messages that allow a reply option of 0, the automatic response function of the System/36 environment can be used to run the equivalent AS/400 command without issuing the specified error message. For more information about using the automatic response function, see the RESPONSE procedure and the // NOHALT OCL statement in this book and in the *System/36 Environment Programming* book.

Table 6-1 shows the following:

- Control commands that are not supported
- ID of the message sent when using the control command
- Replies of 0 that are allowed or not allowed
- Name of the equivalent AS/400 command

Table 6-1 (Page 1 of 3). S/36 Control Command not supported.

Control Command Not Supported	Message ID	Reply of 0 Allowed	Equivalent AS/400 Command
ASSIGN	SYS7260	No	None
CANCEL SESSION	SYS7262	Yes	WRKUSRJOB USER(*ALL) STATUS(*ACTIVE)
CANCEL JOBQ,jobname	SYS7261	Yes	WRKJOBQ JOBQ(QGPL/QBATCH)
CANCEL jobname	SYS7263	Yes	WRKUSRJOB USER(*ALL) STATUS(*ACTIVE)
CHANGE PRTY	SYS7265	Yes	CHGJOB
CHANGE JOBQ	SYS7266	Yes	WRKJOBQ JOBQ(QGPL/QBATCH)
CONSOLE TAKE	SYS7267	Yes	CHGMSGQ MSGQ(QSYSOPR) DLVRY(*BREAK)
CONSOLE GIVE	SYS7268	Yes	CHGMSGQ MSGQ(QSYSOPR) DLVRY(*HOLD)
MODE	SYS7260	No	None

## Control Commands

Table 6-1 (Page 2 of 3). S/36 Control Command not supported.

Control Command Not Supported	Message ID	Reply of 0 Allowed	Equivalent AS/400 Command
POWER	SYS7270	Yes	PWRDWN SYS OPTION(*CNTRLD)
RELEASE JOBQ	SYS7272	Yes	WRKJOBQ JOBQ(QGPL/QBATCH)
REPLY	SYS7273	Yes	DSPMSG MSGQ(*SYSOPR) MSGTYPE(*ALL)
RESTART PRT,,PAGE	SYS5874	No	None
START JOB	SYS7274	Yes	WRKUSRJOB USER(*ALL) STATUS(*ACTIVE)
START JOBQ,jobname	SYS7275	Yes	WRKJOBQ JOBQ(QGPL/QBATCH)
START SERVICE	SYS7260	No	None
START SESSION	SYS7260	No	None
START SYSTEM	SYS7276	Yes	STRSBS
START WORKSTN	SYS7277	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*DEV)
STATUS COMM	SYS7278	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*LIN)
STATUS COMCNFIG	SYS7279	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*LIN)
STATUS LINE	SYS7280	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*LIN)
STATUS SUBSESS	SYS7260	No	None
STATUS SUBSYS	SYS7260	No	None
STATUS APPC	SYS7260	No	None
STATUS MSRJE	SYS7281	Yes	WRKRJESSN
STATUS WORKSTN	SYS7287	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*DEV)
STATUS SYSTASK	SYS7260	No	None
STATUSF WORKSTN	SYS7287	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*DEV)
STOP JOB	SYS7282	Yes	WRKUSRJOB USER(*ALL) STATUS(*ACTIVE)
STOP SESSION	SYS7260	No	None
STOP SERVICE	SYS7260	No	None
STOP SYSTEM	SYS7283	Yes	ENDSBS
STOP WORKSTN	SYS7284	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*DEV)
VARY ON,ctrl ID	SYS7285	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*CTL)
VARY OFF,ctrl ID	SYS7285	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*CTL)

Table 6-1 (Page 3 of 3). S/36 Control Command not supported.

Control Command Not Supported	Message ID	Reply of 0 Allowed	Equivalent AS/400 Command
VARY ON,,line number	SYS7286	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*LIN)
VARY OFF,,line number	SYS7286	Yes	WRKCFGSTS CFGD(*ALL) CFGTYPE(*LIN)

Table 6-1 on page 6-1 describes each of the control commands. The description contains the following information:

- What the command can do and who can use it
- The syntax format of the command. For a discussion of the rules used to describe the syntax format, see “Conventions Used for Describing Syntax Formats” on page 1-3.
- Description of the commands parameters
- One or more examples of how to use the control command

For information on operating the AS/400 system, see the *System Operation* book for your system unit.

---

## ASSIGN Control Command

Not supported.

---

## CANCEL Control Command

Use the CANCEL control command to:

- Cancel one or all of your spool file entries.
- Cancel all of your spool file entries with a specific forms number.
- Cancel your jobs on the job queue.

If you control one or more printers, you can also use the CANCEL control command to:

- Cancel one or all spool file entries for a specific printer under your control.
- Cancel all spool file entries for all printers under your control.
- Cancel all spool file entries with a specific forms number for all printers under your control.
- Cancel all spool file entries with a specific user ID for all printers under your control.

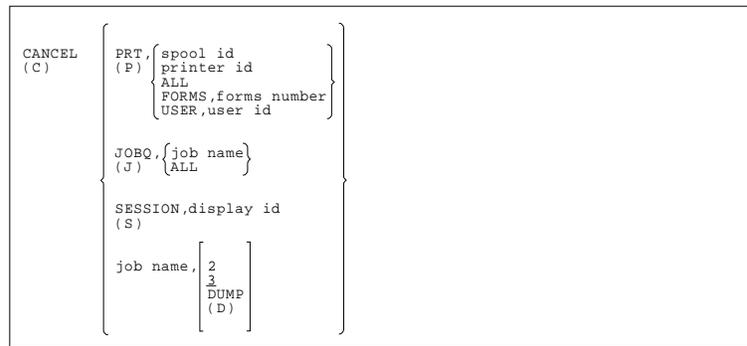
**Note:** If you have \*JOBCTL authority, you can control all printers attached to your system.

If you have \*JOBCTL authority, you can also use the CANCEL control command from any type of display station to do the following:

- Cancel any of the following on the spool file:
  - One entry
  - All entries

## CANCEL

- All entries for a specific printer
- All entries with a specific forms number
- All entries with a specific user ID
- Cancel any one or all jobs on the job queue.



**PRT or P** This value specifies cancelation of one or more entries from the spool file.

**spool id** This value specifies the 6-character ID of the spool file entry canceled. Use the STATUS PRT command to determine the spool file IDs.

**printer id** This value specifies the 2-character ID of a printer for which all spool file entries are canceled. You must have \*JOBCTL authority to use this parameter.

**ALL** This value specifies cancelation of all System/36 environment spool entries and all AS/400 system spool entries to which you are authorized. You must have \*JOBCTL authority to use this parameter.

**FORMS** This value specifies cancelation of all spool file entries with the specified forms number. If you do not have \*JOBCTL authority, this parameter cancels only your spool file entries with the specified forms number. If you have \*JOBCTL authority, this parameter cancels all entries with the specified forms number for all the printers you control.

### forms number

This value specifies the 1- to 4-character forms number of the printout form used for the canceled spool file entries. Use the STATUS PRT control command to determine the forms number used for each spool file entry.

**USER** This value specifies cancelation of all spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries are canceled. If you specify a user ID other than your own, all spool file entries with the specified user ID are canceled for all printers. You must have \*JOBCTL authority to specify a user ID other than your own.

**user id** This value specifies the 8-character user ID for all canceled spool file entries. Use the STATUS PRT control command to determine the user ID.

**JOBQ or J**

This value specifies cancelation of one or more jobs from the job queue.

**job name** This parameter is not supported. Use the Work with Job Queue (WRKJOBQ) command to perform this function.

**ALL** This value specifies cancellation of all job queue entries for the System/36 environment job queue. Both the System/36 environment and the AS/400 job queue entries are canceled.

**Note:** This command affects only the AS/400 job queue QBATCH in library QGPL.

**SESSION or S**

Not supported. Use the End Job (ENDJOB) command or the End option on the Work with User Jobs (WRKUSRJOB) command to end a user's interactive job.

**job name** Not supported. Use the ENDJOB command or the End option from the Work with User Jobs (WRKUSRJOB) command to perform this function.

**DUMP or D**

Not supported. Use the Dump Job (DMPJOB) command to perform this function.

**Example 1**

The following examples cancel spool file entry SP0010:

```
CANCEL PRT,SP0010
```

or:

```
C P,SP0010
```

**Example 2**

The following examples cancel all spool file entries with user ID AA120199:

```
CANCEL PRT,USER,AA120199
```

or:

```
C P,USER,AA120199
```

**Example 3**

The following example cancels all jobs on the job queue:

```
CANCEL JOBQ,ALL
```

---

## CHANGE Control Command

Use the CHANGE control command to change the following:

- The number of copies printed for one or all of your spool file entries
- The number of copies printed for all of your spool file entries with a specific forms number
- The defer status for one of your spool file entries
- The forms number used for one or all of your spool file entries
- The forms number used for all of your spool file entries with a specific forms number
- The printer used for one or all of your spool file entries
- The printer used for all of your spool file entries with a specific forms number

## CHANGE

- The number of copies, forms number, or printer for your user ID

If you control one or more printers, you can also use the CHANGE command to change the following:

- The number of copies printed for a spool file entry for a printer you control
- The number of copies printed for all spool file entries with a specific forms number for all printers you control
- The number of copies printed for all spool file entries with a specific user ID for all printers you control
- The defer status of a spool file entry for a printer you control
- The forms number used for a spool file entry for a printer you control
- The forms number used for all spool file entries with a specific forms number for all printers you control
- The forms number used for all spool file entries with a specific user ID for all printers you control
- The printer used for a spool file entry for a printer you control
- The printer used for all spool file entries for a printer you control
- The printer used for all spool file entries with a specific forms number for all printers you control
- The printer used for all spool file entries with a specific user ID for all printers you control
- The position of a spool file entry for a printer you control
- The number of separator pages placed between the printed output for a printer you control

**Note:** If you have \*JOBCTL authority, you control all printers.

If you have \*JOBCTL authority, you can also use the CHANGE command to change the number of job queue jobs the system keeps active.

CHANGE (G)	{	COPIES,copies,{	spool id	}
			FORMS,forms number	}
			USER,user id	}
		DEFER,{	YES ,spool id	}
			NO	}
		FORMS,forms number,{	spool id	}
			FORMS,forms number	}
			USER,user id	}
		ID,new printer id,{	old printer id	}
			spool id	}
			FORMS,forms number	}
			USER,user id	}
		PRT,spool id,{	spool id1	}
			(P)	}
		PRTY,{	HIGH	}
			NORMAL	}
			printer id	}
			system printer	}
		SEP,{	0	}
			1	}
			2	}
			3	}
			printer id	}
			system printer	}
		JOBQ,job name,{	job name1	}
			(J)	}
		JOBS,{	JOBQ	}
			JOB queue priority	}
			number of jobs	}

59020343-3

- COPIES** This value specifies a change in the number of copies of printed output for one or more spool file entries.
- copies** This value specifies the number of copies to print for the spool file entry or entries. Specify any number from 1 through 255.
- spool id** This value specifies the 6-character ID of a spool file entry. Use the STATUS PRT control command to determine the spool file IDs.
- FORMS** This value specifies a change in the number of copies of all spool file entries with the specified forms number. If you do not have \*JOBCTL authority, only your spool file entries with the specified forms number change. If you have \*JOBCTL authority, all entries with the specified forms number change for all printers.
- forms number**  
This value specifies the 1- to 4-character forms number of the spool file entries for which the number of copies changes. Use the STATUS PRT control command to determine which form to use for each spool file entry.
- USER** This value specifies a change in the number of copies of all spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries change. If you specify a user ID other than your own, all spool file entries with the specified user ID change. You must have \*JOBCTL authority to specify a user ID other than your own.
- user id** This value specifies the 8-character user ID of the spool file entries for which the number of copies changes. Use the STATUS PRT control command to determine the user ID.
- DEFER** This value specifies that the defer status changes for a particular spool file entry.
- YES** This value indicates that the spool file entry should not start printing until the program completes the spool file entry.
- NO** This value indicates that the spool file entry can start printing before the program completes the spool file entry. If you do not specify a parameter, NO is assumed.
- spool id** This value specifies the 6-character ID of a spool file entry. The spool file entry must not yet be completed by the program creating it. Use the STATUS PRT control command to determine the spool file IDs.
- Note:** You cannot change the defer status when your entry is printing.
- FORMS** This parameter specifies a change in the forms number used for one or more spool file entries. Use the STATUS WRT control command to determine which forms are currently in use by the printer. The STATUS PRT control command identifies which forms to use for each spool file entry.
- forms number**  
This value specifies the new forms number of the printout form used. Enter from 1 to 4 characters.

## CHANGE

**spool id** This value specifies the 6-character ID of a spool file entry. The spool file entry must not be printing. Use the STATUS PRT command to determine the spool file IDs.

**FORMS** This value specifies a change in the printout form for all spool file entries with the specified forms number. If you do not have \*JOBCTL authority, only your spool file entries with the specified forms number change. If you have \*JOBCTL authority, all entries with the specified forms number change.

**forms number**

This value specifies the current 1- to 4-character forms number of the spool file entries for which the printout form changes.

**USER** This value specifies a change in the printout form for all spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries change. If you specify a user ID other than your own, all spool file entries with the specified user ID change for all printers that you control. You must have \*JOBCTL authority to specify a user ID other than your own.

**user id** This value specifies the 8-character user ID of the spool file entries for which the printout form changes. Use the STATUS PRT control command to determine the user ID.

**ID** This parameter specifies a change of printer used for one or more spool file entries.

**Note:** If you direct your printed output to another printer with different printer characteristics, printing or programming errors can result, or your output may not print properly (word processing jobs may not complete successfully).

**new printer id**

This value specifies the 2-character ID of the new printer to use.

**old printer id**

This value specifies the 2-character ID of the current printer. All spool file entries printed using this printer use the new printer. If a spool file entry currently prints on the old printer, it continues printing on that printer. You must have \*JOBCTL authority to enter this parameter.

**spool id** This value specifies the 6-character ID of a spool file entry. Use the STATUS PRT control command to determine the spool file IDs. If you specify a spool ID in the third parameter position, that spool file entry uses the new printer. The spool file entry must not be printing.

**FORMS** This value specifies use of the new printer by all spool file entries with the specified forms number. If you do not have \*JOBCTL authority, only your spool file entries with the specified forms number change. If you have \*JOBCTL authority, all entries with the specified forms number change.

- forms number** This value specifies the 1- to 4-character forms number of the spool file entries for which the printer changes. Use the STATUS PRT control command to determine the forms number.
- USER** This value specifies use of the new printer by all spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries change. If you specify a user ID other than your own, all spool file entries with the specified user ID change. You must have \*JOBCTL authority to specify a user ID other than your own.
- user id** This value specifies the 8-character user ID of the spool file entries for which the printer changes. Use the STATUS PRT control command to determine the user ID.
- PRT or P** This parameter specifies change in the position of an entry on the spool file.
- spool id** This value specifies the 6-character ID of the spool file entry that moves to the top of the output queue. Use the STATUS PRT control command to determine the spool file IDs.
- spool id1** Not supported. Use the Change Spooled File Attributes (CHGSPLFA) command with the output priority (OUTPTY) parameter to perform this function.
- PRTY** Not supported. Use the Change Job (CHGJOB) command with the run priority (RUNPTY) parameter to change the processing priority of the spool writer.
- SEP** This parameter specifies a change in the number of separator pages printed between spool file entries. You can specify 0, 1, 2, or 3 separator pages. If no parameter is specified, 0 (zero) is assumed. Separator pages are used to separate the output for different print files. You must have \*JOBCTL authority to enter this parameter.
- printer id** This value specifies the 2-character ID of the printer whose number of separator pages changes. If you do not specify a parameter, the system printer is assumed.
- Note:** You can only change the number of separator pages when the spool writer is active.
- JOBQ or J** Not supported. Use the Work with Job Queues (WRKJOBQ) command to move a job to the top of the job queue by changing the job queue priority to 0, or the CHGJOB command to change the priority of a job queue job.
- JOBS** This parameter specifies a change in the number of jobs run at once from the job queue, or the number of jobs run at once from a job queue priority. You can enter this parameter only if you have \*JOBCTL authority.
- JOBQ** This value specifies a change in the number of jobs run at once from the job queue.

## CHANGE

### job queue priority

This value specifies a change in the number of jobs run at once from the given job queue priority.

**Note:** The values this command enters remain in effect until the next CHANGE JOBS, START JOBQ, or STOP JOBQ command is entered.

### number of jobs

This value specifies the number of jobs run from the job queue or specified job queue priority. The number of jobs run from the job queue is from 0 through 1000. The number of jobs run from a specific priority is 0 through 99.

**Note:** This command affects only the AS/400 job queue QBATCH in library QGPL.

The System/36 environment uses the following values to map the job queue priority to the OS/400 priority:

System/36 Environment Job Queue Priority	OS/400 Priority
5	3
4	4
3	5
2	6
1	7
0	8

### Example 1

The following examples change spool file entry SP0011 so that five copies of its output print:

```
CHANGE COPIES,5,SP0011
```

or:

```
G COPIES,5,SP0011
```

### Example 2

The following examples, if entered by the system operator, change the forms number to 6987 for all spool file entries with current forms number 1324:

```
CHANGE FORMS,6987,FORMS,1324
```

or:

```
G FORMS,6987,FORMS,1324
```

### Example 3

The following examples, if entered by the operator controlling printer P3, cause no separator pages to print between jobs at printer P3:

```
CHANGE SEP,0,P3
```

or:

```
G SEP,,P3
```

---

## CONSOLE Control Command

Not supported. Use the Change Message Queue (CHGMSGQ) command to perform this function.

---

## HOLD Control Command

Use the HOLD control command to prevent one of your entries on the spool file from printing.

If you have \*JOBCTL authority, you can also use the HOLD control command to do the following:

- Prevent a spool file entry from printing
- Prevent all spool file entries from printing for a specific printer
- Prevent all spool file entries from printing for all printers

If a HOLD PRT control command specifies a spool file entry that is printing, the command interrupts the printing of that entry and begins printing the next entry on the spool file. You can print the held entry by issuing a RELEASE control command.

<pre>HOLD (H)   { PRT, { spool id   (P) { printer id       ALL   }   }   { JOBQ, job name   (J)   }</pre>
---

59020345-0

**PRT or P** This parameter specifies the holding of one or more entries on the spool file.

**spool id** This value specifies the 6-character ID of the spool file entry to hold. Use the STATUS PRT control command to determine the spool file IDs.

**printer id** This value specifies the 2-character ID of a printer for which all entries on the output queue are held. Only the entries currently on the output queue are held. Entries placed on the output queue after the HOLD PRT control command are not held. You must have \*JOBCTL authority to use this parameter.

**ALL** This value specifies that all entries on the output queue for all the printers are held. Only the entries currently on the output queue are held. Any entries you place on the output queue after the HOLD command processes are not held. You must have \*JOBCTL authority to use this parameter.

**JOBQ or J**

Not supported. Use the Work with Job Queues (WRKJOBQ) command to hold a job queue job.

**Example 1**

The following examples hold entry SP0012 on the spool file:

```
HOLD PRT,SP0012
```

or:

```
H P,SP0012
```

**Example 2**

The following example, entered by the operator controlling display station P3, holds the spool file entries for printer P3:

```
H P,P3
```

---

**INFOMSG Control Command**

Use the INFOMSG control command to specify whether you want informational messages to appear.

The INFOMSG command can be used to prevent /\* OCL statements from rolling up on the SYSIN display.

```
INFOMSG [ YES ]
          [ NO ]
```

59020346-0

**YES** This parameter specifies that any informational messages appear. If you do not specify a parameter, YES is assumed.

**NO** This parameter specifies that any informational messages the /\* statement issues do not appear.

**Example**

The following example prevents display of any informational messages sent to the display station:

```
INFOMSG NO
```

---

**JOBQ Control Command**

Use the JOBQ control command to place a job on the job queue.

Enter the JOBQ control command from the command display. Do not exceed 120 characters.

```
JOBQ (J) [ job queue priority, ] [ library name ] ,procedure name
          [ 3, ] [ current library ]
          [ ,parm1 ,parm2 ,... ]
```

59020347-1

**job queue priority**

This parameter specifies job queue priority (the job's order of processing from the job queue). The job queue priority is any number from 0 through 5. When the system chooses the next job to run, it considers jobs with higher priority numbers before jobs with lower priority numbers. For example, it considers all jobs with a job queue priority of 5 before any other jobs in the job queue. The system considers jobs with the same job queue priority in the order they were placed in the job queue. The system considers jobs with a job queue priority of 0 last.

Priority level 0 is not automatically stopped at IPL on the AS/400 system. To set up priority level 0 to run similar to priority level 0 on System/36, you must add the Change Job Queue Element (CHGJOBQE) command to your start up procedure (#STARTUP1 or #STARTUP2).

An example of the command is as follows:

```
CHGJOBQE SBSD(QSYS/QBATCH) JOBQ(QGPL/QBATCH) MAXPTY8(0)
```

This example sets the minimum number of jobs active for System/36 environment priority level 0 to 0 and stops priority level 0 jobs from running.

If you do not specify a parameter, the system assigns a System/36 environment job queue priority based on the job priority (JOBPTY) attribute of the job description associated with the System/36 environment session that put the job on the job queue. Do not specify the comma shown in this parameter.

The System/36 environment uses the following values to map the job queue priority to the OS/400 system priority:

System/36 Environment Job Queue Priority	OS/400 System Priority
5	3
4	4
3	5
2	6
1	7
0	8

**library name**

This parameter specifies the library for the job. The system searches the specified library, the system library, and then the user's library list for the procedures, load members, message members, and display formats that this job uses. If you do not specify a parameter, the current library is assumed.

**procedure name**

This parameter specifies the procedure that defines the job to place on the job queue.

**parm1,parm2,. . .**

This parameter specifies any parameters required by the procedure. You can specify from 1 through 64 parameters. See "INCLUDE OCL Statement" on page 5-49 for more information about how you can specify parameters.

## MENU

**Note:** The JOBQ control command submits jobs to the AS/400 job queue QBATCH in library QGPL.

### Example 1

The following example, entered by the operator, places the PAYROLL procedure, which is located in library PAYLIB, on the job queue with a priority of 3:

```
JOBQ PAYLIB,PAYROLL,FILEA,FILEB
```

### Example 2

The following example places the PAYROLL procedure, which is located in library PAYLIB, on the job queue with a job queue priority of 5:

```
JOBQ 5,PAYLIB,PAYROLL,FILEA,FILEB
```

---

## MENU Control Command

Use the MENU control command to display a menu. Menus allow you to select procedures or control commands by a number, rather than having to enter that procedure or control command. If you try to display a double-byte character set (DBCS) menu at a non-DBCS display station, an error message appears.

You can use the MENU control command to display the OS/400 menus (\*MENU objects) in addition to the System/36 environment menus. For more information, refer to the following:

- The *ADTS/400: Screen Design Aid for the System/36 Environment* book
- The Create Menu (CRTMNU) command in the *CL Reference* book
- The *System/36 Environment Programming* book

Enter the MENU command from a command display.

```
MENU menu name, [library name]
```

59020348-0

### menu name

This parameter specifies the user menu that appears.

### library name

This parameter specifies the library containing the menu. If you do not specify a library name, the system searches for the menu, in the following order:

1. The current library, if one exists
2. The system library (#LIBRARY)
3. The library list

The library you specify becomes the session library.

### Example 1

The following example displays menu M12, which is in the current library:

```
MENU M12
```

### Example 2

The following example displays menu PAYROL, which is in the library MYLIB:

```
MENU PAYROL,MYLIB
```

---

## MODE Control Command

Not supported. On System/36, standby mode allows a signed-on display to be acquired by a program.

On the AS/400 system, programs can acquire only display stations that are powered on but signed off. The ability to acquire signed-on display stations is not supported.

**Note:** Because of this difference, System/36 applications that rely on system verification of user authorization must now do their own security checking to prevent unauthorized users from running the program.

Also, acquired display stations cannot use System Request. An acquired display station session has no local data area (LDA).

---

## MSG Control Command

The MSG control command can display a message sent to you or your display station.

The MSG control command can also send a message to any of the following:

- A selected display station
- A selected user
- All display stations
- A selected personal computer location
- The system operator
- A user on another system

Messages sent are placed in one of the following types of OS/400 message queues:

- Each display station has an associated work station message queue. The name of this message queue is the same as the OS/400 device name, not the System/36 environment display ID.
- Each user on the system has an associated user profile message queue.
- Each personal computer location has an associated message queue. The name of this message queue is the same as the personal computer location name.
- The system operator message queue is QSYSOPR.

Messages sent to a message queue are retained on the queue until deleted by a user.

**Note:** This is different from the System/36, on which all user messages displayed or older than 7 days at IPL are deleted, and all work station messages displayed or at IPL are deleted.

The message waiting light is set on when a message is sent to a message queue that is in notify delivery mode and the severity of the message is greater than the severity code filter for the message queue. Once messages are displayed, the message waiting light will be reset.

**Note:** Messages sent using the MSG control statement use the following severities:

- 00 for (user id, address), (list id, qualifier), or group name
- 99 for ALL
- 80 for all other messages

The following search order is used to find the message queue to which the message is to be sent:

1. (user id, address), or (list id, qualifier)
2. ALL
3. blank
4. System/36 environment display ID
5. User ID
6. OS/400 display station, personal computer location, or user-defined (The message sent is placed in the first message queue in your library list that matches the specified name.)
7. group name

If a user is not signed on when a message is sent, and the user has a user profile, the message is placed in the user profile message queue. The message can then be displayed by the user when he signs on.

See the *System Operation* book for more information on message queues. See the *System/36 Environment Programming* book for more information on messages in the System/36 environment.

```

MSG [ [ System/36 environment display id ] ,message text ]
    [ user id
      pc location
      user-defined
      (user id,address)
      (list id, qualifier)
      ALL
      group name ] ]

```

S9020677-0

If no parameters are specified, the messages in the work station message queue and user profile message queue are shown. Assuming there are messages in both queues, two message displays are shown. The first display shows the messages in the work station message queue. The second display shows the messages in the your user profile message queue. If one of two message queues contains no messages, only the queue with messages is shown. If neither queue contains messages, a display is shown that says no messages are available.

If the first parameter is not specified and the second parameter is specified, the message is sent to the system operator message queue (QSYSOPR) in library QSYS.

#### **System/36 environment display id**

This specifies the System/36 environment 2-character display ID to which the message is to be sent to. The message is sent to the work station message queue of the associated OS/400 display station. To find out the System/36 environment display ID and its associated OS/400 display station use the Display System/36 (DSPS36) command.

#### **OS/400 display station**

This specifies the 1- to 8-character OS/400 display station to which the message is to be sent to.

**user-id** This specifies the 1- to 8-character user profile name that identifies the user profile message queue of the user to whom the message is to be sent. If you specify QSYSOPR, the message will be sent to the user profile message queue associated with the QSYSOPR user profile.

#### **pc location**

This specifies the 1- to 8-character personal computer location name to which the message is to be sent.

#### **user-defined**

This specifies the 1- to 8-character name of a user-defined message queue to which the message is to be sent. The user-defined message queue must be in a library in your library list.

#### **(user-id,address)**

This specifies the 1- to 8-character user id and the 1- to 8-character address that together identify the person in the distribution network to whom the message is to be sent to. The user id and address must already exist in the system distribution directory. See the *SNA Distribution Services* book for more information on the system distribution directory.

#### **(list-id,qualifier)**

This specifies the 1- to 8-character distribution list ID and the 1- to 8-character distribution list ID qualifier that together identify a list of users enrolled in the system distribution directory for SNA Distributions

## OFF

Services to whom the message is to be sent. See the *SNA Distribution Services* book for more information on the system distribution directory.

**ALL** This specifies that the message is sent to all display stations and personal computer locations on the system. Messages sent to all display stations are always sent in break mode.

### group name

This specifies the 1- to 8-character name of a distribution list ID. The System/36 environment will use your system name as the distribution list ID qualifier. The distribution list ID and the distribution list ID qualifier together identify a list of users enrolled in the system distribution directory for SNA Distributions Services to whom the message is to be sent. See the *SNA Distribution Services* book for more information on the system distribution directory.

### message text

This specifies the contents of the message to be sent. Up to 75 characters can be entered. DBCS messages are always sent, but may only display correctly on DBCS capable display stations.

### Example 1

The following example, entered by the display station operator, informs the system operator that the display station operator is preparing to submit a job that requires diskette BFILE:

```
MSG ,PLEASE PUT DISKETTE BFILE INTO SLOT 1
```

### Example 2

The following example, entered by the system operator, notifies the operator at display station W4 that he cannot enter any more jobs:

```
MSG W4,PLEASE DO NOT SUBMIT ANY MORE JOBS
```

### Example 3

The following example sends a message to all display stations on the system:

```
MSG ALL,PLEASE END YOUR JOBS, I NEED A DEDICATED SYSTEM
```

---

## OFF Control Command

The OFF command is mapped to the SIGNOFF CL command. If the command default for the SIGNOFF CL command on your system is LOG(\*NOLIST), a job log is not produced for the job when the OFF command is processed. If the command default for the SIGNOFF CL command on your system is LOG(\*LIST), a job log is produced for the job when OFF command is processed. The command defaults for the SIGNOFF CL command are changed using the Change Command Default (CHGCMDDFT) CL command. See the *CL Reference* book for more information on the SIGNOFF and CHGCMDDFT CL commands.

Enter the OFF command from a command display.



S9020351-0

Use DROP and HOLD only for remote display stations on switched communications lines. These parameters are ignored for other display stations.

**DROP** This parameter specifies that the communications session for this display is no longer needed.

**HOLD** This parameter specifies holding of the communications session for the display. A Sign On display appears.

### Example

The following example signs your display station off the system:

OFF

**Note:** On System/36, DROP was the default. The OS/400 default is to use the attribute defined in the device description for the display station. Use the Change Device Description (Display) (CRTDEV DSP) command specifying \*YES for the drop line at sign-off (DROP) parameter.

---

## POWER Control Command

Not supported. Use the Power Down System (PWRDWNSYS) command to power down the system.

**Note:** The PWRDWNSYS command does not check for active jobs before powering down the system.

---

## PRTY Control Command

Not supported. Use the Change Job (CHGJOB) command to change the priority of your job. Use the Work with User Jobs (WRKUSRJOB) command to change the priority of another user's job.

---

## RELEASE Control Command

Use the RELEASE control command to release one of your entries on the spool file for printing.

If you have \*JOBCTL authority, you can also use the RELEASE control command to do the following:

- Release one or all entries on the spool file for a specific printer.
- Release all held entries on the spool file for all printers.

<pre> RELEASE { PRT, { spool id (L)      (P) { printer id }           ALL           }         }         {         JOBQ,job name         (J)         } </pre>
--

59020354-0

**PRT or P** This parameter specifies release of one or more spool file entries.

**spool id** This value specifies the 6-character ID of the spool file entry to release. Use the STATUS PRT control command to determine the spool file IDs.

## RESTART

**printer id** This value specifies the 2-character ID of a printer. The system releases all the spool file entries printed on the specified printer and releases only the entries currently held on the output queue. Any held entries you place on the output queue after the RELEASE command processes are not released. You must have \*JOBCTL authority to use this parameter.

**ALL** This value specifies release of all spool file entries for all printers. The system releases only the entries currently held on the output queue, and does not release any held entries you place on the output queue after the RELEASE command processes. You must have \*JOBCTL authority to use this parameter.

### **JOBQ or J**

Not supported. Use the Work with Job Queues (WRKJOBQ) command to release jobs held on the job queue.

### **Example 1**

The following example releases entry SP0013 on the spool file:

```
RELEASE PRT,SP0013
```

### **Example 2**

The following example releases all the entries on the spool file:

```
L P,ALL
```

### **Example 3**

The following example, entered by the operator controlling display station P3, releases the spool file entries for printer P3:

```
L P,P3
```

---

## REPLY Control Command

Not supported. Use the Display Message (DSPMSG) command to display and respond to messages.

---

## RESTART Control Command

Use the RESTART control command to restart printing any one of your entries that is in WTR status on the spool file.

If you have sufficient authority, such as \*JOBCTL authority, you may use the RESTART control command to restart any entry that is in WTR status on the spool file. See the *System/36 Environment Programming* book for other authorities that would allow you to restart any entry that is in WTR status on the spool file.

You can restart the printing at the beginning of the specified entry or from the top of a specified page of that entry.

**Notes:**

1. If you restart a spool file entry on its starting page and separator pages are being used, separator pages are reprinted.
2. On an Advanced Function Printing (AFP), several spool file entries may be buffered together waiting to be printed. The spool file entry that is presently in WTR status for the AFP printer may not be the spool file entry that currently is printing. Therefore, when using the RESTART control command with a AFP printer, be careful not to restart the wrong spool file entry.

RESTART (T)	PRT, (P)	[ spool id printer id system printer ]	, [ page number PAGE ]
----------------	-------------	--	---------------------------

S9020687-1

**PRT or P** This parameter specifies to restart the printing of a spool file entry.

**spool id** This value specifies the 6-character ID of the spool file entry to restart. Use the STATUS PRT control command to determine the spool IDs of the entries on the spool file.

**printer id** This value specifies the 2-character ID of the printer on which the entry in WTR status is restarted. If no printer ID or spool file ID is specified, the system printer is assumed. Use the STATUS WRT control command to determine appropriate printer IDs.

**page number**

This value specifies the number of the page on which printing is to be restarted. Printing is restarted at the top of the specified page. The specified page number must be within the range defined by the PAGERANGE attribute of the spool file entry. If a page number is not specified, printing restarts at the beginning page of the spool file entry. Use the STATUS PRT control command to determine the values defined by the PAGERANGE attribute.

**PAGE** Not supported. Specify the page number where printing is to restart.

**Example 1**

If spool file entry SP0012 is in WTR status, the following examples restart printing of the spool file entry at the top of page 6:

```
RESTART PRT,SP0012,6
```

```
T P,SP0012,6
```

**Example 2**

If a spool file entry assigned to printer P2 is in WTR status, the following examples restart printing of the spool file entry at the top of page 99:

```
RESTART PRT,P2,99
```

```
T P,P2,99
```

# START

## Example 3

If a spool file entry assigned to the system printer is in WTR status, the following examples restart printing of the spool file entry at the top of the beginning page:

```
RESTART PRT
```

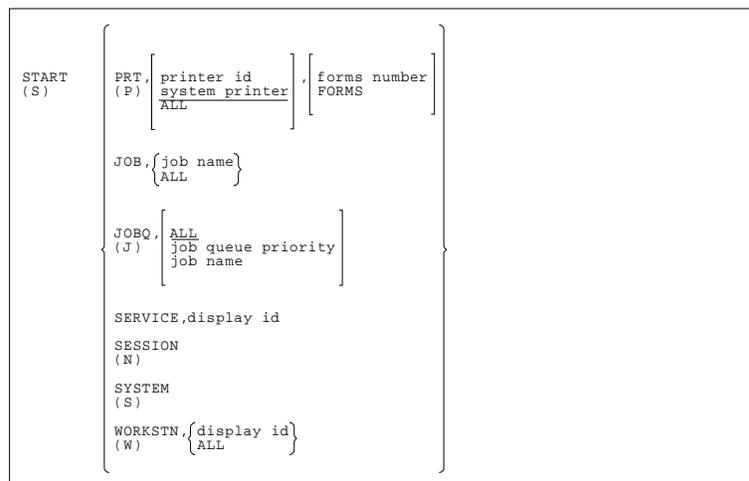
```
T P
```

---

## START Control Command

If you control one or more printers, use the START control command to start the printing of:

- All spool file entries for a specific printer or for all printers
- All spool file entries with a specified forms number for a specific printer
- All spool file entries for a specific printer so those entries using the same forms print together



**PRT or P** This parameter specifies starting a spooling writer for a printer. Starting the spooling writer allows printing of spool file entries when the printer is available. Use the PRT parameter to start the printing of entries on the output queue after IPL if the prompt for starting the spooling writers was answered with an N during IPL. You can use the PRT parameter to start the spooling writer after you enter the STOP PRT command.

If you do not enter a second parameter, the system printer is assumed. If you do not enter a third parameter, the spooling writer prints the available entries according to their position in the spool file.

**printer id** This value specifies the 2-character ID of the printer for which the spooling writer is to start. If you do not specify a parameter, the system printer is assumed. Use the STATUS WRT control command to determine whether the spooling writer for a particular printer is started or stopped.

**ALL** This value specifies starting of the spooling writers for all printers. You must have \*JOBCTL authority to use this parameter.

**forms number**

This value specifies the forms number used by the spooling writers. Enter from 1 to 4 characters. Only entries with the specified forms numbers print.

**FORMS**

This value specifies that the spooling writer prints all available spool file entries that require the forms currently used in the printer, regardless of the entry's position in the spool file. After all such entries print, the operator controlling the printer receives a prompt to change the forms, and prints all entries using the next group of forms. Using FORMS reduces the number of operator messages to change the forms.

**JOB**

Not supported. Use the Work with User Jobs (WRKUSRJOB) command to display all jobs, and select the release option to start a stopped job.

**JOBQ or J**

This parameter specifies that all jobs on the job queue, or those jobs with the specified job queue priority, can process. When the job queue empties, job running begins automatically when you place one or more jobs on the job queue. You must have \*JOBCTL authority as well as \*ALL authority to the QINTER and QBATCH subsystem descriptions to use this parameter.

**ALL**

This value specifies that all jobs running at a priority that maps to a System/36 environment priority on the job queue can run. If no parameter is specified, ALL is assumed. ALL sets the maximum number of active jobs to 1 for all System/36 environment priorities. This parameter value resets any values set by a CHANGE JOBS command.

**Note:** This command affects only the AS/400 job queue QBATCH in library QGPL.

The System/36 environment uses the following values to map the job queue priority to the OS/400 priority:

System/36 Environment Job Queue Priority	OS/400 Priority
5	3
4	4
3	5
2	6
1	7
0	8

**job queue priority**

This value starts jobs on the job queue with the specified job queue priority. Enter any number from 0 through 5. This sets the maximum number of active jobs for a priority to 1. This parameter value resets any values set by a CHANGE JOBS control command.

**job name**

This value is not supported. Use the Change Job (CHGJOB) command to start a job on the job queue.

**SERVICE** Not supported.

## STATUS

### **SESSION or N**

Not supported.

### **SYSTEM or S**

Not supported. Use the Start Subsystem (STRSBS) command to perform this function.

### **WORKSTN or W**

Not supported. Use the Work with Configuration Status (WRKCFGSTS) command to start a previously stopped work station.

### **Example 1**

The following example starts running jobs from the job queue after a STOP JOBQ control command stopped the initiation of jobs from the job queue:

```
START JOBQ
```

### **Example 2**

The following example, entered by the operator controlling display station P3, starts printer P3:

```
S P,P3
```

---

## STATUS Control Command

Use the STATUS control command to display the status of the following:

- Your own or another display station session
- Jobs you have placed on the job queue
- Entries on the spool file
- Display stations, printers, display station pass-through and distributed host command facility (DHCF) devices, the diskette drive, and tape drives
- Spool writers

**Note:** You can enter the STATUS PRT command and press F4 (Prompt) to see a prompt display that allows you to complete parameters.

STATUS (D)	SESSION, [display id] (S)
	COMM, [line number] (C)
	COMCNFIG, [line number] (H)
	LINE, [line number] (L)
	JOBQ, [job name] [priority] (J)
	PRT, [printer id], [ALL] (P)
	SUBSESS (N)
	SUBSYS (I)
	APPC, [location name] (A)
	MSRJE, [location name] (M)
	WORKSTN, [display id] [printer id] I1 T1 T2 TC (W)
	WRT, [printer id]
	SYSTASK, [job name] [system task id] (T)
	MESSAGE, [display id] (G)
	USERS, [job name] (U) ALERT

RV2P182-1

**SESSION or S**

This parameter displays information about the display station session. If you do not specify a first parameter, SESSION is assumed. If you do not specify a display ID, the status of your display station appears.

**display id**

This value specifies the 2-character ID of the display station for which status information appears.

**COMM or C**

Not supported. You can use the Work with Configuration Status (WRKCFGSTS) command to display the line descriptions on your system.

**COMCNFIG or H**

Not supported. You can use the Work with Configuration Status (WRKCFGSTS) command to display the line descriptions on your system.

**LINE or L**

Not supported. You can use the Work with Configuration Status (WRKCFGSTS) command to display the line descriptions on your system.

**JOBQ or J**

This parameter displays entries on the job queue:

**job name** This is supported for System/36 compatibility only. The value is not used.

**priority** This is supported for System/36 compatibility only. The value is not used.

**Note:** Only entries on the System/36 environment job queue QBATCH in library QGPL are displayed.

The System/36 environment uses the following values to map the job queue priority to the OS/400 priority:

System/36 Environment Job Queue Priority	OS/400 Priority
5	3
4	4
3	5
2	6
1	7
0	8

**PRT or P** This parameter displays information about spool file entries. (After entering the STATUS PRT command, you can press F4 (Prompt) to see the prompt display for the Work with Spooled Files (WRKSPLF) command. This prompt display allows you to fill in additional selection information.) If you specify a printer ID for the second parameter, only the information for spool file entries for that printer appear. If you do not specify a printer ID, the information about spool file entries for all printers appear. If you specify ALL for the third parameter, the information about all spool file entries appears. If you do not specify ALL, only the information about spool file entries for the user running the command appears.

**printer ID**

This value specifies the 2-character ID of the printer for which spool file entry information is to appear.

**ALL**

This value specifies that information about all spool file entries on the system is to appear.

Another way to display information about all spool file entries or to display information about spool file entries for a specific user is to press F4 (Prompt) after you enter the STATUS PRT command. The prompt for the Work with Spool Files (WRKSPLF) command appears. On this prompt, you can enter \*ALL or a user ID in the User field on the command prompt.

**SUBSESS or N**

The Display SSP-ICF Session Status (D N) command is not supported. Similar information is available using the Work with Configuration Status (WRKCFGSTS), Work with Active Jobs (WRKACTJOB), and Display Job (DSPJOB) commands.

- To get information equivalent to the JOB NAME information, use the WRKCFGSTS command, specifying \*DEV for the type (CFGTYPE) parameter. For every job on the system that has an active session with a device, the display shows a detail line indicating the job name. The detail line is at the following levels:
  - Device description level for single session devices
  - Mode level for APPC device descriptions
- For more information about the job using the device, you can do the following:
  1. Select option 5 (Work with job) from the Work with Configuration Status display.
  2. Use the Work with Active Job (WRKACTJOB) command to display a list of active jobs. Then select option 5 (Work with) for the job name that was shown on the Work with Configuration Status display.
  3. Use the Display Job (DSPJOB) command, specifying the job name taken from the Work Configuration Status display.
- For communication information, select option 14 (Display open files) on the Work with Job menu. All file types are shown on this display. For communications, refer to the file type labeled ICF.
  - The SYMID is found under the **Member/Device** heading.
  - The operation for any session can be determined through the **Record Format** heading. The record format can be a system-supplied format, for example \$\$SEND, or a user-defined DDS format. The operation performed is based on the record format being used.

Open file information can also be displayed by specifying \*OPNF for the option (OPTION) parameter on the DSPJOB command.

- For communication status information, select option 17 (Display communications status) on the Work with Job menu. Communications status information can also be displayed by specifying \*CMNSTS for the option (OPTION) parameter on the DSPJOB command.
- The value equivalent to the STATE of the operation can be determined from the Work with Active Job display. The STATE is found under the **Status** heading and it has the following meaning:
  - RUN** Running
  - MSGW** Waiting on a message
  - ICFW** Waiting on a communications operation
  - ICFA** Active communications operation
- The value equivalent to the PROGRAM NAME can be found on the Work with Active Job display under the **Function** heading.
- The type of the job can also be determined from the Work with Active Job display under the **Type** heading. The following is a list of job types:
  - INT** Interactive job

**BCH** Batch job  
**BCI** Batch Immediate job  
**EVK** Evoked job

- The LOCATION NAME can be determined by using the WRKCFGSTS command, by specifying \*DEV for the type (CFGTYPE) parameter. On the Work with Configuration Status display, select option 8 (Work with description. On the Work with Mode Description display, select option 5 (Display). The remote location is displayed on the Display Device Description display.

### **SUBSYS or I**

The Display Subsystem Status (D I) command is not supported. Similar information is available from the Work with Configuration Status (WRKCFGSTS) command.

- The status of a communications configuration (STATUS and COMMUNICATING) can be determined using the Work with Configuration Status (WRKCFGSTS) command, specifying \*DEV for the type (CFGTYPE) parameter.
  - A status of ACTIVE is equivalent to the System/36 status of ENABLED and COMMUNICATING.
  - A status of VARIED ON PENDING is equivalent to the System/36 status of ENABLING.
  - A status of VARIED ON is equivalent to the System/36 status of ENABLED and NOT COMMUNICATING.
- The number of evoked and acquired sessions can also be determined using the WRKCFGSTS command specifying \*DEV for the CFGTYPE parameter. For every job on the system that has an active session to a device, there is a detail line indicating the job name. The detail line is at the following levels:
  - device description level for single session devices
  - mode level for APPC device descriptions
- More information about the configuration itself can be found in the configuration descriptions.
  - To determine the remote location name (LOCATION NAME) and the communications type (CONFIG TYPE) use the WRKCFGSTS command specifying \*DEV for the type (CFGTYPE) parameter. Use F14 (Work with device description), and select option 5 (Display device description).
  - To determine the line being used, use the WRKCFGSTS command, specifying \*LIN for the type (CFGTYPE) parameter. Select F14 (Work with Lines), and select option 5 (Display line description). This will give you the resource name associated with the line.
- The CONFIG NAME does not apply to the AS/400 system.

### **APPC or A**

Not supported. You can use the Display Mode Status (DSPMODSTS) command or the Work with Configuration Status (WRKCFGSTS) command to do similar functions.

**MSRJE or M**

Not supported. Use the Work with Remote Job Entry Session (WRKRJESSN) command to perform this function.

**WORKSTN or W**

Not supported. You can use the Work with Configuration Status (WRKCFGSTS) command, specifying CFGTYPE(\*DEV) CFGD(\*ALL), to display the status of all devices on your system.

**WRT** This parameter displays the status of the spool writers.

**printer id** Not supported.

**SYSTASK or T**

Not supported.

**MESSAGE or G**

This parameter displays the status of the spool writers. You can then select the option to display messages for an active spool writer.

**USERS or U**

This parameter displays the status of all user jobs running on the system. If a job name is specified, it is ignored.

**ALERT** This parameter displays the status of active and enabled alert locations.

**Example**

The following commands display the information about spool file entries for the user running the commands. The spool file entries are for all printers on the system.

```
STATUS PRT
```

or:

```
D P
```

The following commands display the information about spool file entries for the user running the commands. The spool file entries are only for printer P1.

```
STATUS PRT,P1
```

or:

```
D P,P1
```

The following commands display the information about spool file entries for all users on the system. The spool file entries are for all printers on the system.

```
STATUS PRT,,ALL
```

or:

```
D P,,ALL
```

The following commands display the information about spool file entries for all users on the system. The spool file entries are only for printer P2.

```
STATUS PRT,P2,ALL
```

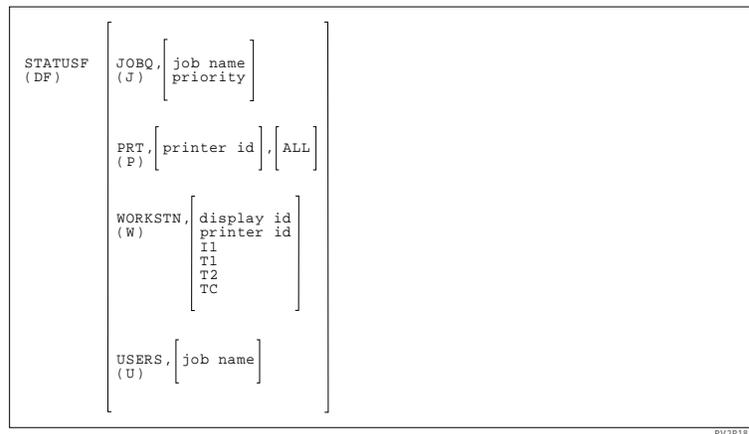
or:

D P,P2,ALL

## STATUSF Control Command

Use the STATUSF control command to display the status of the following:

- Jobs on the job queue
- Entries on the spool file
- Display stations, printers, display station pass-through and distributed host command facility (DHCF) devices, the diskette drive, and tape drives
- A selected job or all jobs running on the system



### JOBQ or J

This parameter displays entries on the job queue.

**job name** This is supported for System/36 compatibility only. The value is not used.

**priority** This is supported for System/36 compatibility only. The value is not used.

**Note:** Only entries in QBATCH in library QGPL on the System/36 environment job queue are displayed.

The System/36 environment uses the following values to map the job queue priority to the OS/400 priority:

System/36 Environment Job Queue Priority	OS/400 Priority
5	3
4	4
3	5
2	6
1	7
0	8

**PRT or P** This parameter displays information about spool file entries. (After entering the STATUSF PRT command, you can press F4 (Prompt) to see the prompt display for the Work with Spooled Files (WRKSPLF) command. This prompt display allows you to fill in additional selection information.) If you specify a printer ID for the second parameter, only

the information for spool file entries for that printer appear. If you do not specify a printer ID, the information about spool file entries for all printers appear. If you specify ALL for the third parameter, the information about all spool file entries appears. If you do not specify ALL, only the information about spool file entries for the user running the command appears.

**printer ID**

This value specifies the 2-character ID of the printer for which spool file entry information is to appear.

**ALL**

This value specifies that information about all spool file entries on the system is to appear.

Another way to display information about all spool file entries or to display information about spool file entries for a specific user is to press F4 (Prompt) after you enter the STATUS PRT command. The prompt for the Work with Spool Files (WRKSPLF) command appears. On this prompt, you can enter \*ALL or a user ID in the User field on the command prompt.

**WORKSTN or W**

Not supported. You can use the Work with Configuration Status (WRKCFGSTS) command, specifying CFGTYPE(\*DEV) CFGD(\*ALL), to display the status of all devices on your system.

**USERS or U**

This parameter displays the status of all user jobs running on the system. If a job name is specified, it is ignored.

**Example**

The following commands display the information about spool file entries for the user running the commands. The spool file entries are for all printers on the system.

```
STATUSF PRT
```

or:

```
DF P
```

The following commands display the information about spool entries for the user running the commands. The spool file entries are only for printer P1.

```
STATUSF PRT,P1
```

or:

```
DF P,P1
```

The following commands display the information about spool file entries for all users on the system. The spool file entries are for all printers on the system.

```
STATUSF PRT,,ALL
```

or:

```
DF P,,ALL
```

## STOP

The following commands display the information about spool file entries for all users on the system. The spool file entries are only for printer P2.

```
STATUSF PRT,P2,ALL
```

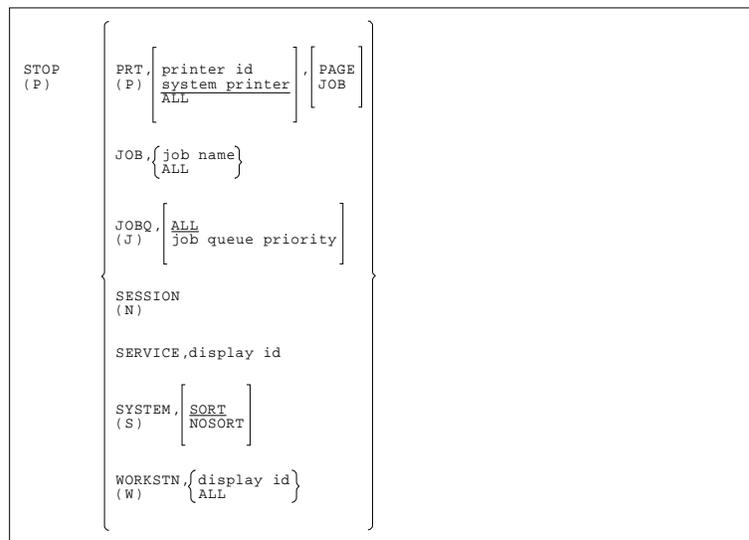
or:

```
DF P,P2,ALL
```

---

## STOP Control Command

If you have \*JOBCTL authority, use the STOP control command to stop the printing of all entries from the spool file for a specified printer, or for all printers that you control. You can stop printing immediately, at the end of the currently printing page, or at the end of the currently printing spool file entry.



**PRT or P** This parameter stops the printing of entries on the spool file. If an entry is printing, and you do not specify a third parameter, printing stops immediately. Resume the printing by entering the START PRT control command.

**printer id** This value specifies the 2-character ID of the printer for which the spooling writer stops. If you do not specify a printer ID, the system printer is assumed.

**ALL** This value specifies that printing stops for all printers.

**PAGE** This value stops the spooling writer when it has completed printing the current page.

**JOB** This value stops the spooling writer when it completes printing the current copy of the spool file entry.

**JOB** Not supported. Use the Work with User Jobs (WRKUSRJOB) command and select the option to hold a job.

**JOBQ or J**

This parameter prevents starting all jobs, or jobs with the specified job queue priority from the job queue. The job already running continues until complete.

**Note:** This command resets to 0 any previously set values of the CHANGE JOBS command. You must have \*JOBCTL authority as well as \*ALL authority to the QINTER and QBATCH subsystem descriptions to use this parameter.

**ALL** This value prevents all jobs on the job queue from starting. ALL sets the maximum number of active jobs to 0 for all System/36 environment job queue priorities. This parameter value resets any values set by a CHANGE JOBS command.

**job queue priority**

This value specifies a job queue priority number from 0 through 5. It prevents all jobs with the specified job queue priority from starting.

This sets the maximum number of active jobs for a priority to 0. This parameter value resets any value set by a CHANGE JOBS command.

**Note:** This command affects only the AS/400 job queue QBATCH in library QGPL.

The System/36 environment uses the following values to map the job queue priority to the OS/400 priority:

System/36 Environment Job Queue Priority	OS/400 Priority
5	3
4	4
3	5
2	6
1	7
0	8

**SESSION or N**

Not supported.

**SERVICE** Not supported.

**SYSTEM or S**

Not supported. Use the End Subsystem (ENDSBS) command with the OPTION \*CTRLD parameter to perform this function.

**WORKSTN or W**

Not supported. Use the Work with Configuration Status (WRKCFGSTS) command and select option 5 (Work with job) to perform this function.

### Example

The following examples, entered by the operator controlling printer P3, stop printing on printer P3:

```
STOP PRT,P3
```

or:

```
P P,P3
```

## TIME Control Command

Use the TIME control command to display the time of day and the system date. The time is based on the time specified by the system operator during IPL, and appears in the following base format:

```
hh:mm:ss
```

**hh** This value specifies the hours.  
**mm** This value specifies the minutes.  
**ss** This value specifies the seconds.

The system date appears in the system date format:

```
mmddy, dmmyy, or yymmdd
```

**mm** This value specifies the month.  
**dd** This value specifies the day.  
**yy** This value specifies the year.

When the time advances from 23:59:59 to 00:00:00, the system date advances. The System/36 environment supports the optional date-and-time separators under QTIMSEP and QDATESEP. The optional date-and-time separators are blank ( ), comma (,), and period (.).



59020362-0

The TIME command has no parameters.

### Example

Typing the following displays the time of day and the system date:

```
TIME
```

## VARY Control Command

If you have \*JOBCTL authority, you can enter the VARY control command at any type of display station to change the status of the following devices from online to offline, and from offline to online:

- Display stations
- Printers
- System printer

- Diskette drive
- Tape drives

Operators or programs cannot use offline devices. You cannot use the VARY command to take offline a device that is allocated to a program or signed on.

**Note:** On System/36, when a display or printer is varied on, the line and controller are also varied on. On the AS/400 system, varying a device online does not vary the associated line and controller online. To vary on a device, you must ensure that the line and controller are varied on.

You can use the AS/400 Work with Configuration Status (WRKCFGSTS) command to determine the current status of devices on the system.

```

VARY      {ON } , { display id
(V)       {OFF} , { printer id
                PRT
                (P)
                I1
                T1
                T2
                TC
                controller id
                ,line number
                controller id,line number

```

59020363-2

**ON** This parameter causes the specified device to be placed online.

**OFF** This parameter causes the specified device to be placed offline.

#### **display id**

This parameter specifies the System/36 environment 2-character ID of a display station to place online or offline. Use the Display System/36 (DSPTS36) command to determine the display station IDs.

To place a display station offline, you must sign it off.

**printer id** This parameter specifies the System/36 environment 2-character ID of a printer to place online or offline. Use the DSPTS36 command to determine the printer IDs.

The spooling writers for a printer must be stopped or complete to place that printer offline. Use the AS/400 WRKCFGSTS command to determine whether a spooling writer is stopped or complete.

**PRT or P** This parameter specifies placing the system printer online or offline.

**I1** This parameter specifies placing the diskette drive online or offline.

**T1** This parameter specifies placing of tape drive 1 online or offline.

**T2** This parameter specifies placing of tape drive 2 online or offline.

**TC** This parameter specifies placing of the tape cartridge drive online or offline.

#### **controller id**

This parameter is not supported. Use the Vary Configuration (VRYCFG) command or the Work with Configuration Status (WRKCFGSTS) command to perform this function.

## VARY

### **,line number**

This parameter is not supported. Use the VRYCFG or the WRKCFGSTS command to perform this function.

### **controller id,line number**

This parameter is not supported. Use the VRYCFG or the WRKCFGSTS command to perform this function.

### **Example**

The following example places device W2 online:

```
VARY ON,W2
```

---

## Appendix A. System/36 Environment Utility Programs

This appendix shows the System/36 environment utility programs you can use instead of the System/36 environment procedures. The utility programs are in alphabetical order by utility, with a description of the functions you can perform using that utility, and the operation control language (OCL) and utility control statements you can use to perform that function.

---

### Making Your Own Procedures Using System/36 Environment Utility Programs

This section describes why you may want to make your own procedures from the System/36 environment utility programs. By making your own procedures based on the statements shown, you can specify OCL statements to change how the procedures run. For example, you can specify the following information:

- Different job or job step dates, using the DATE OCL statement. See “DATE OCL Statement” on page 5-19 for more information about changing the date.
- Different disk file retention types, using the FILE OCL statement. See “FILE OCL Statement (for Disk Files)” on page 5-25 for more information about disk file retention types.
- Special printer directions, using the PRINTER OCL statement. See “PRINTER OCL Statement” on page 5-68 for more information about specifying printer information.

This section shows an example based on the statements you can use instead of the supplied procedure.

#### Example

The following example creates a version of the LISTLIBR procedure that prints output with a priority of 0. This priority causes the spool file to hold the output until it is specifically released for printing. You could, for example, run this sample procedure and then run the COPYPRT procedure to display the output.

The name of the procedure is LSAMPLE. To test it, enter it into a library procedure member. See “List Library Members and Information (LISTLIBR Procedure)” on page A-51 and “PRINTER OCL Statement” on page 5-68 for information about the statements used in this procedure.

## System/36 Environment Utility Programs

```
* LSAMPLE procedure
// LOAD $MAINT
// PRINTER NAME-$SYSLIST,SPOOL=YES,PRIORITY=0
// RUN
// COPY TO-PRINT,NAME-?1R?,LIBRARY-?2'S'?,FROM-?3'?CLIB'?
// END
// RETURN
*
* Sample procedure (based on LISTLIBR procedure)
*
* Parameter      Entry      Meaning
* -----
* Parameter 1:  member name  Name of member to list
*               DIR          List library directory
*               ALL          List all members
* -----
* Parameter 2:  S            List source members (default)
*               P            List procedure members
*               O            List load members
*               R            List subroutine members
*               ALL          List all member types
* -----
* Parameter 3:  library name  Default is the current library
```

The statements in the LSAMPLE procedure indicate the following:

### **asterisk (\*)**

In the first column, indicates that the statement is a comment statement. Any information following the asterisk is not processed.

**LOAD** This indicates loading of the \$MAINT utility program.

**PRINTER** This indicates how the system list output printed by \$MAINT changes. The NAME-\$SYSLIST parameter indicates that the statement modifies the system list output. The SPOOL=YES parameter indicates spooling of the system list output. The PRIORITY=0 parameter causes holding of the output on the spool file. That is, the output is not printed until you release it (by using the RELEASE command, for example).

**RUN** This indicates that the \$MAINT program begins running. The \$MAINT program then reads the COPY and END statements.

**COPY** This indicates listing of the specified library member, or directory (if you specify DIR).

**END** This indicates the end of the \$MAINT utility control statements.

**RETURN** This indicates that procedure ends. The system does not process the following comment statements.

To run the LSAMPLE procedure, first use the source entry utility (SEU) to enter it into a library procedure member named LSAMPLE. You do not have to enter the comment (\*) statements. After you enter the procedure, run it. For example, to list a library source member named TEST from a library named MYLIB, enter:

```
LSAMPLE TEST,,MYLIB
```

To list the directory of the library MYLIB, enter:

```
LSAMPLE DIR,ALL,MYLIB
```

---

## \$ARSP Utility

The \$ARSP utility program lets you change the automatic response values and the severity levels in a message load member (RESPONSE procedure).

```

// LOAD $ARSP
// RUN

// RESPONSE SOURCE=source member name [ ,LIBRARY={library name} ] [ ,ALERTS={YES} ]
//                                     [ ,#LIBRARY ] [ ,NO ]

// END

```

S9020364-1

## Changing Automatic Response Values (RESPONSE Procedure)

See "RESPONSE Procedure" on page 4-196 for more information.

### Example

The following example applies the automatic responses contained in the library source member named AUTORESP (which is stored in a library named MYLIB):

```

// LOAD $ARSP
// RUN
// RESPONSE SOURCE=AUTORESP,LIBRARY=MYLIB
// END

```

## Changing Alert Indicators (SETALERT Procedure)

Not supported. See "SETALERT Procedure" on page 4-237 for more information.

---

## \$BICR Utility

The \$BICR utility program lets you:

- List a basic data exchange or I-exchange file on diskette (LISTFILE procedure).
- Convert a basic data exchange or I-exchange diskette file to a sequential or an indexed disk file (TRANSFER procedure).
- Add a diskette file that is in basic data exchange or I-exchange format to an existing sequential disk file (TRANSFER procedure).
- Convert a disk file to a basic data exchange or I-exchange diskette file (TRANSFER procedure).

## Listing Diskette Files (LISTFILE Procedure)

See "LISTFILE Procedure" on page 4-142 for more information.

```

// LOAD $BICR
// FILE NAME=COPYIN,LABEL=input file name,UNIT-I1
// RUN

// DISPLAY [ FROM=first record [ ,TO=last record ] ]

// END

```

S9020365-0

## \$BICR (TRANSFER)

Not all FILE OCL parameters are shown. For information about the FILE OCL statement, see “FILE OCL Statement (for Diskette Files)” on page 5-33 and “FILE OCL Statement (for Disk Files)” on page 5-25.

The DISPLAY statement causes the file to be listed on the system list device. Use the STATUS SESSION command to determine your system list device.

### FROM-first record

This specifies the first record displayed. If you do not specify FROM, the file is listed, beginning with the first record.

### TO-last record

This specifies the last record displayed. If you do not specify TO, the file is listed, ending with the last record. The maximum value you can specify is 8000000.

### Example

The following example lists a basic data exchange diskette file named BASICDAT:

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-BASICDAT,UNIT-I1
// RUN
// DISPLAY
// END
```

## Copying Files (TRANSFER Procedure)

See “TRANSFER Procedure” on page 4-289 for more information.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-input file name,UNIT- $\left\{ \begin{array}{l} F1 \\ I1 \end{array} \right\}$ 
// FILE NAME-COPYO,LABEL-output file name,UNIT- $\left\{ \begin{array}{l} F1 \\ I1 \end{array} \right\}$ 
// RUN
// TRANSFER  $\left[ \text{ADD-} \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \left[ ,\text{KEYLEN-key length,KEYLOC-key location} \right]$ 
//
//  $\left[ ,\text{FORMAT-} \left\{ \begin{array}{l} \text{EXCHANGE} \\ \text{IFORMAT} \end{array} \right\} \right]$ 
// END
```

59020366-0

### Notes:

1. The ADD-YES parameter is not supported on the AS/400 system when transferring files to diskette.
2. You cannot create a diskette file that contains a period in the label using \$BICR.

Not all FILE OCL parameters are shown. For information about the FILE OCL statement, see “FILE OCL Statement (for Diskette Files)” on page 5-33 and “FILE OCL Statement (for Disk Files)” on page 5-25.

**Example 1**

The following example creates a disk sequential file named FILE2 from a diskette basic data exchange or I-exchange file named FILE2:

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-FILE2,UNIT-I1
// FILE NAME-COPYO,LABEL-FILE2,UNIT-F1
// RUN
// TRANSFER
// END
```

**Example 2**

The following example creates an indexed disk file named FILE2 from a basic data exchange file named FILE2. The key is in positions 1 through 5.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-FILE2,UNIT-I1
// FILE NAME-COPYO,LABEL-FILE2,UNIT-F1
// RUN
// TRANSFER ADD-NO,KEYLEN-5,KEYLOC-1
// END
```

**Example 3**

The following example adds a basic data exchange or I-exchange diskette file named FILE1 to an existing disk file named FILE1:

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-FILE1,UNIT-I1
// FILE NAME-COPYO,LABEL-FILE1,UNIT-F1
// RUN
// TRANSFER ADD-YES
// END
```

**Example 4**

The following example creates an I-exchange diskette file named FILE3 on a diskette from a disk file named FILE3. The file is to be saved for 30 days. The volume ID of the diskette is VOL002.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-FILE3,UNIT-F1
// FILE NAME-COPYO,LABEL-FILE3,UNIT-I1,PACK-VOL002,RETAIN-30
// RUN
// TRANSFER FORMAT-IFORMAT
// END
```

---

**\$BMENU Utility**

The \$BMENU utility creates the library members required to display a menu. See "BLDMENU Procedure" on page 4-25 for more information.

## \$BUILD (BUILD)

```
[ LIBRLIBR source member library,load member library,SOURCE,
  menu name##,,REPLACE ]
LIBRLIBR source member library,load member library,SOURCE,
text member name,,REPLACE
REMOVE menu name##,LOAD,load member library
REMOVE text member name,LOAD,load member library

CREATE menu name##,,load member library

[ CREATE text member name,,load member library ]

// LOAD $BMENU
// RUN

// MENU INPMSG-menu name## [ ,MENMSG-text member name ] [ ,REPLACE-{NO}
                                     {YES} ]

                                     [ ,INLIB-{load member library} ] [ ,FREEFORM-{NO}
                                     {YES} ] [ ,IGC-{NO}
                                     {YES} ]

// END

REMOVE text member name,LOAD,load member library
REMOVE menu name##,SOURCE,load member library
REMOVE text member name,SOURCE,load member library ]
```

59020367-0

### Example

This example assumes the following conditions:

- You have created a command text source member named MENU## and an option text source member named MENUUDT. These source members contain System/36 message source statements.
- Both members are stored in a library named MYLIB.

To build a menu named MENU, using the \$BMENU utility, enter the following information:

```
CREATE MENU##,,MYLIB
CREATE MENUUDT,,MYLIB
// LOAD $BMENU
// RUN
// MENU INPMSG-MENU##,MENMSG-MENUUDT,INLIB-MYLIB,
//      REPLACE-NO,FREEFORM-NO
// END
REMOVE MENUUDT,LOAD,MYLIB
```

The library member created by \$BMENU is placed in the specified library called MYLIB.

---

## \$BUILD Utility

Not supported. See "BUILD Procedure" on page 4-29 for more information.

---

## **\$COPY Utility**

The \$COPY utility program lets you:

- Copy disk files (COPYDATA procedure)
- List disk, diskette, tape, or tape cartridge files (LISTDATA and LISTFILE procedures)
- Restore disk files from diskette, tape, or tape cartridge (RESTORE procedure)
- Save disk files on diskette, tape, or tape cartridge (SAVE procedure)

Also shown in this section are \$COPY utility control statements that are supported for compatibility with IBM System/34.

### **Notes:**

1. The \$COPY utility cannot copy deleted records from a delete-capable file to a nondelete-capable file.
2. When copying records from one direct file to another using the \$COPY utility, no operation that excludes records can be performed. Therefore, you cannot perform a copy operation involving a delete-capable input file and a nondelete-capable output file. Also, on the COPYFILE utility control statement, you cannot specify INCLUDE, OMIT, POSITION, CHAR, DELETE, or REORG. You can specify LIMIT.
3. LIMIT on the COPYFILE utility control statement and TO on the SELECT utility control statement cannot be specified in the same \$COPY utility job step.
4. The \$COPY utility can also be used with physical database data files that do not have a System/36 environment direct, indexed, or sequential file organization.
  - When an output disk file is created using such files, it may be given file attributes which differ from those you requested or expected.
  - When such a file is saved, the saved file may not be given the file attributes you requested.

A message allowing you to continue is sent whenever this may occur. See the *System/36 Environment Programming* book for more information on using System/36 environment files and AS/400 files.

## **Copying Disk Files (COPYDATA Procedure)**

See "COPYDATA Procedure" on page 4-50 for more information.

## \$COPY (COPYDATA)

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-copy from file name,UNIT-F1
// FILE NAME-COPYO,LABEL-copy to file name,UNIT-F1
// RUN

// COPYFILE [ OUTPUT- { SAME
                  SEQUENTL
                  (S)
                  INDEXED
                  (I)
                  DIRECT
                  (D) } ] [ , { INCLUDE } - { EQ
                               OMIT          NE
                                           LT
                                           LE
                                           GE } ] [ , POSITION-position ]

[ , CHAR- { 'characters' } ] [ , REORG- { NO
                                         YES } ] [ , RECL-record length ]

[ , LIMIT-maximum records ]

[ // SELECT { RECORD } , FROM-starting value [ , TO-ending value ]
           { KEY
           { PKY } ] ]

[ // KEY   POSITION-key position , LENGTH-key length ]

// END
```

59020369-0

Not all FILE OCL statement parameters are shown. For more information on the FILE statement, see “FILE OCL Statement (for Disk Files)” on page 5-25.

**Note:** If you specify DISP-OLD in the FILE OCL statement containing NAME-COPYO, the COPYO file may not be deleted from disk if you select option 2 or 3 in response to an error message during the \$COPY job step or job. In this case, the COPYO file may contain invalid records.

### Differences from the Procedure Parameters

For the COPYFILE statement:

#### CHAR-'characters'

This lets you specify up to 30 characters of data. The data must begin and end with apostrophes ('). You can also specify apostrophes in the character string by entering two apostrophes. For example, enter o'clock as: 'o'clock'.

If your system recognizes double-byte character set (DBCS) data, the character string may contain a mixture of DBCS and non-DBCS characters. You must enclose a DBCS character string with a shift-out (hex 0E) character and a shift-in (hex 0F) character. If there is a shift-out character in the first position of the total character string, that shift-out character is not included for comparison. Likewise, if there is a shift-in character in the last position of the string, that shift-in character is not included for comparison. All other shift-out and shift-in characters are part of the comparison string. Each DBCS character occupies 2 characters in the character string while each shift-out and shift-in character occupies 1 character.

**Note:** If a string of non-DBCS characters has an EBCDIC value equivalent to the DBCS characters you have specified in the comparison string and it lies in the specified position, the record containing the string of non-DBCS characters can be included or omitted (depending on what you specified earlier for the

INCLUDE or OMIT keyword). Take care when dealing with files whose records are not all in the same format.

**Xdddd...dd**

This can also specify the comparison characters. This form specifies hexadecimal data. You may specify up to 15 bytes, where 2 digits (dd) represent a hexadecimal byte. The hexadecimal digits are 0 through 9 and A through F. The X must precede the hexadecimal digits.

The SELECT statement lets you further define the records to be copied.

**RECORD** This specifies that a portion of the file is to be copied. When you specify RECORD, the FROM parameter and the TO parameter (if you use TO) must specify relative record numbers.

**FROM-starting value**

This specifies the relative record number of the first record to be copied. For example, if the first record to be copied is the fifth record in the file, you should specify FROM-5. If only one record is to be copied, the FROM and TO parameters must specify the same relative record number.

The FROM value must be less than or equal to the TO value.

**TO-ending value**

This specifies the relative record number of the last record to be copied. For example, if the last record to be copied is the fifteenth record in the file, you should specify TO-15. If only one record is to be copied, the FROM and TO parameters must specify the same relative record number. If you do not specify an ending value, the file is copied through the last record in the file. You cannot specify the TO parameter unless you also specify the FROM parameter.

The TO value must be greater than or equal to the FROM value.

**KEY or PKY**

This specifies that a portion of an indexed file is to be copied. PKY must be specified if the indexed file contains packed keys. If the file contains packed keys, you may specify up to 239 numeric characters. If the file contains regular keys, you may specify up to 120 characters.

**FROM-starting value**

This specifies the key (or the beginning characters of the key) of the first record to be copied. You must enclose the characters in apostrophes ('). If none of the keys in the file begin with the specified characters, the record with the next higher key is the first record copied. For example, if you specify FROM-'NAME1', the first key beginning with NAME1 or a larger value is the key of the first record copied.

The FROM value must be less than or equal to the TO value. If only one record is to be copied, the FROM and TO parameters must specify the same key.

## \$COPY (COPYDATA)

### TO-ending value

This specifies the key (or the beginning characters of the key) of the last record to be copied. You must enclose the ending value in apostrophes (').

If none of the keys in the file begin with the specified characters, the record with the next lower key is the last record copied. For example, if you specify TO-'34', the last key beginning with 34 is the key of the last record copied. If no keys begin with 34, the last key that begins with a value smaller than 34 is the key of the last record copied.

The TO value must be greater than or equal to the FROM value.

If you do not specify the TO parameter, \$COPY uses the last key in the index as the TO key. If only one record is to be copied, the FROM and TO parameters must specify the same key. You cannot specify the TO parameter without specifying the FROM parameter.

### Example 1

The following example copies FILE1 and creates an exact copy of it, FILE2:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE2
// RUN
// COPYFILE
// END
```

**Example 2**

The following example copies a file named FILE1, which is an indexed file, and creates a new file named FILE3. The new file is an indexed file and the key is in positions 5 through 24.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE3
// RUN
// COPYFILE OUTPUT-INDEXED
// KEY POSITION-5,LENGTH-20
// END
```

**Example 3**

The following example uses the SELECT statement to copy from FILE1 to the new file, FILE7, only those records that have the relative record numbers 2, 3, and 4:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE7
// RUN
// COPYFILE
// SELECT RECORD,FROM-2,TO-4
// END
```

**Listing \$COPY Files (LISTDATA/LISTFILE Procedures)**

See "LISTDATA Procedure" on page 4-137 or "LISTFILE Procedure" on page 4-142 for more information.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-input file name,UNIT- $\left\{ \begin{array}{l} F1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ 

// RUN

// COPYFILE OUTPUT- $\left\{ \begin{array}{l} CHAR \\ HEX \\ PARTHEX \\ CRT \end{array} \right\}$   $\left[ \begin{array}{l} ,INCLUDE- $\left\{ \begin{array}{l} EQ \\ NE \\ LT \\ LE \\ GT \\ GE \end{array} \right\}$  \\ ,OMIT \end{array} \right]$   $\left[ ,POSITION-position \right]$ 

 $\left[ ,CHAR-'characters' \right]$   $\left[ ,REORG- $\left\{ \begin{array}{l} NO \\ YES \end{array} \right\}$  \right]$   $\left[ ,RECL-record length \right]$ 

 $\left[ ,LIMIT-maximum records \right]$   $\left[ ,IGC- $\left\{ \begin{array}{l} YES \\ NO \end{array} \right\}$  \right]$ 

 $\left[ \begin{array}{l} // SELECT  $\left\{ \begin{array}{l} RECORD \\ KEY \\ PKY \end{array} \right\}$  ,FROM-starting value  $\left[ ,TO-ending value \right]$  \end{array} \right]$ 

// END
```

59020370-2

Not all file statement parameters are shown in the previous diagram. For more information about the FILE statement, see "FILE OCL Statement (for Disk Files)" on page 5-25, "FILE OCL Statement (for Diskette Files)" on page 5-33, and "FILE OCL Statement (for Tape Files)" on page 5-37.

## **Differences from the Procedure Parameters**

For the COPYFILE statement:

### **CHAR-'characters'**

This lets you specify up to 30 characters of data. The data must begin and end with apostrophes ('). You can also specify apostrophes in the character string by entering two apostrophes. For example, enter o'clock as: 'o'clock'.

If your system recognizes DBCS data, the character string may contain a mixture of DBCS and non-DBCS characters. Enclose a DBCS character string in a shift-out (hex 0E) character and a shift-in (hex 0F) character. If there is a shift-out character in the first position of the total character string, that shift-out character is not included for comparison. Likewise, if there is a shift-in character in the last position of the string, that shift-in character is not included for comparison. All other shift-out and shift-in characters are part of the comparison string. Each DBCS character occupies 2 characters in the character string while each shift-out and shift-in character occupies one character.

**Note:** If a string of non-DBCS characters has an EBCDIC value equivalent to the DBCS characters you have specified in the comparison string, and it lies in the specified position, the record containing the string of DBCS characters can be included or omitted (depending on what you specified earlier for the INCLUDE or OMIT keyword). Take care when dealing with files whose records are not all in the same format.

### **CHAR-Xdddd...dd**

This specifies hexadecimal data. You may specify up to 15 bytes, where 2 digits (dd) represent a hexadecimal byte. The hexadecimal digits are 0 through 9 and A through F. The X must come before the hexadecimal digits.

**IGC** The parameter is supported for compatibility. If IGC is specified, the value specified for it (YES or NO) is syntax checked, but otherwise ignored. The system determines whether the file can contain DBCS data for printing. The \$COPY utility cannot display DBCS data at a display station.

For the SELECT utility control statement:

**RECORD** This specifies that a portion of the file is to be listed. When you specify RECORD, the FROM parameter and the TO parameter (if you use TO) must specify relative record numbers.

#### **FROM-starting value**

This specifies the relative record number of the first record to be listed. For example, if the first record to be listed is the fifth record in the file, you should specify FROM-5. If only one record is to be listed, the FROM and TO parameters must specify the same relative record number. The FROM value must be less than or equal to the TO value.

#### **TO-ending value**

This specifies the relative number of the last record to be listed. For example, if the last record to be listed is the fifteenth record in the file, you should specify TO-15. If only

one record is to be listed, the FROM and TO parameters must specify the same relative record number. You cannot specify the TO unless you also specify the FROM parameter. If you do not specify an ending value, the file is listed through the last record in the file. The TO value must be greater than or equal to the FROM value.

**KEY or PKY**

This specifies that a specified portion of an indexed file is to be listed. PKY must be specified if the indexed file contains packed keys. If the file contains packed keys, you may specify up to 239 numeric characters. If the file contains regular keys, you may specify up to 120 characters.

**FROM-starting value**

This specifies the key (or the beginning characters of the key) of the first record to be listed. You must enclose the starting value in apostrophes (''). If none of the keys in the file begins with the specified characters, the record with the next higher key is the first record to be listed. For example, if you specify FROM-'15', the first key beginning with 15 or more will be the key of the first record to be listed. If only one record is to be listed, the FROM and TO parameters must specify the same key. The FROM value must be less than or equal to the TO value.

**TO-ending value**

This specifies the key (or the beginning characters of the key) of the last record to be listed. You must enclose the ending value in apostrophes (''). If none of the keys in the file begins with the specified characters, the record with the next lower key is the last record listed. For example, if you specify TO-'34', the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record listed. You cannot specify the TO parameter unless you also specify the FROM parameter.

If you do not specify the TO parameter, \$COPY uses the last key in the index as the TO key. If only one record is to be listed, the FROM and TO parameters must specify the same key. The TO value must be greater than or equal to the FROM value.

**Example 1**

The following example prints only the printable characters in all the records of the file named FILE1:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

## \$COPY (RESTORE)

### Example 2

The following example prints only records 5 through 10 in the disk file named FILE1. Both the characters in the records and their hexadecimal representations are printed.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// RUN
// COPYFILE OUTPUT-HEX
// SELECT RECORD,FROM-5,TO-10
// END
```

### Example 3

The following example displays all records of a tape file called FILE1. The tape is mounted on tape drive 2.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-T2,LABEL-FILE1
// RUN
// COPYFILE OUTPUT-CRT
// END
```

### Example 4

The following example displays the contents of a tape cartridge file named DATAFILE on the system list device:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-TC,LABEL-DATAFILE
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

## Restoring the Network Resource Directory (RESTNRD Procedure)

Not supported. See "RESTNRD Procedure" on page 4-205 for more information.

## Restoring Files (RESTORE Procedure)

The following example restores all previously saved files of a set:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT- $\left\{ \begin{array}{l} I1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ ,LABEL-set name
// FILE NAME-COPYO,UNIT-F1
// RUN
// COPYALL TO-F1  $\left[ \begin{array}{l} ,STRTLABL-starting \text{ file name} \\ ,STRTDATB-starting \text{ file date} \end{array} \right]$ 
// END
```

59020371-1

The following example restores a single previously saved file:

```

// LOAD $COPY
// FILE NAME-COPYIN,UNIT- $\left\{ \begin{array}{l} I1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ ,LABEL-file name

// FILE NAME-COPYO,UNIT-F1,LABEL-file name
// RUN
// COPYFILE  $\left[ \begin{array}{l} \text{OUTPUT-} \left\{ \begin{array}{l} \text{SAME} \\ \text{SEQUENTL} \\ \text{(S)} \\ \text{INDEXED} \\ \text{(I)} \\ \text{DIRECT} \\ \text{(D)} \end{array} \right\} \left[ \begin{array}{l} \left\{ \text{INCLUDE} \right\} - \left\{ \begin{array}{l} \text{EQ} \\ \text{NE} \\ \text{LT} \\ \text{LE} \\ \text{GT} \\ \text{GE} \end{array} \right\} \end{array} \right] \left[ \text{,POSITION-position} \right] \\ \\ \left[ \text{,CHAR-} \left\{ \text{characters} \right\} \right] \left[ \text{,REORG-} \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \left[ \text{,RECL-record length} \right] \\ \\ \left[ \text{,LIMIT-maximum records} \right] \end{array} \right]$ 

 $\left[ \begin{array}{l} // \text{ SELECT } \left\{ \begin{array}{l} \text{RECORD} \\ \text{KEY} \\ \text{PKY} \end{array} \right\} \text{,FROM-starting value} \left[ \text{,TO-ending value} \right] \end{array} \right]$ 

 $\left[ \begin{array}{l} // \text{ KEY } \text{ POSITION-key position,LENGTH-key length} \end{array} \right]$ 

// END

```

59020372-1

Not all FILE statement parameters are shown. For more information about the FILE OCL statement, see “FILE OCL Statement (for Disk Files)” on page 5-25, “FILE OCL Statement (for Diskette Files)” on page 5-33, and “FILE OCL Statement (for Tape Files)” on page 5-37.

### Differences from the Procedure Parameters

If you specify T1, T2, or TC for the UNIT parameter in the FILE statement, you must specify RECFM-FB, RECL-256, and BLKL-24576.

For the COPYFILE statement:

#### CHAR-'characters'

This specifies up to 30 characters of data. The data must begin and end with apostrophes ('). You can also specify apostrophes in the character string by entering two apostrophes. For example, enter o'clock as: 'o'clock'.

If your system recognizes DBCS data, the character string may contain a mixture of DBCS and non-DBCS characters. You must enclose a DBCS character string with a shift-out (hex 0E) character and a shift-in (hex 0F) character. If a shift-out character appears in the first position of the total character string, it is not included for comparison. If a shift-in character appears in the last position of the string, that shift-in character is not included for comparison. All other shift-out and shift-in characters are part of the comparison string. Each DBCS character occupies 2 characters in the character string while each shift-out and shift-in character occupies one character.

**Note:** If a string of non-DBCS characters has an EBCDIC value equivalent to the DBCS characters you have specified in the comparison string and it lies in the specified position, the record containing the string of non-DBCS characters can be included or omitted (depending on what you specified earlier for the

## **\$COPY (RESTORE)**

INCLUDE or OMIT keyword). Take care when dealing with files whose records are not all in the same format.

### **CHAR-Xdddd...dd**

This specifies hexadecimal data. You may specify up to 15 bytes, where 2 digits (dd) represent a hexadecimal byte. The hexadecimal digits are 0 through 9 and A through F. The X must come before the hexadecimal digits.

### **LIMIT-maximum records**

This specifies the total number of records to be copied into the restored file. You may enter any number greater than 0.

**Note:** The system cannot create a file that contains more than 16 777 200 records.

The SELECT statement lets you further select which records you want to copy.

**RECORD** This specifies that a portion of the file is to be restored. When you specify RECORD, the FROM parameter and the TO parameter (if you use TO) must specify relative record numbers.

### **FROM-starting value**

This specifies the relative record number of the first record to be restored. For example, if the first record to be restored is the fifth record in the file, you must specify FROM-5. If only one record is to be restored, the FROM and TO parameters must specify the same relative record number. The FROM value must be less than or equal to the TO value.

### **TO-ending value**

This specifies the relative record number of the last record to be restored. For example, if the last record to be restored is the fifteenth record in the file, you must specify TO-15. If you do not specify the TO parameter, \$COPY uses the last record in the file as the TO record. If only one record is to be restored, the FROM and TO parameters must specify the same relative record number.

The TO value must be greater than or equal to the FROM value. You cannot specify the TO parameter unless you specify the FROM parameter.

### **KEY or PKY**

This specifies that a portion of an indexed file is to be restored. You must specify PKY if the indexed file contains packed keys. If the file contains packed keys, you may specify up to 239 numeric characters. If the file contains regular keys, you may specify up to 120 characters.

When you use SELECT KEY or PKY to select records from an indexed file, all records whose keys are within the specified limits are restored to the output file. The output records are in key sequence only if the file was saved in key sequence. For information on ordering the records of an indexed file in key sequence, see "SAVE Procedure" on page 4-220.

### **FROM-starting value**

This specifies the key (or the beginning characters of the key) of the first record to be restored. You must enclose the starting value in apostrophes ('). If none of the keys in the

file begins with the specified characters, the record with the next higher key is the first record restored. For example, if you specify FROM-'15', the first key beginning with 15 or more is the key of the first record restored.

The FROM value must be less than or equal to the TO value. If only one record is to be restored, the FROM and TO parameters must specify the same key.

**TO-ending value**

This specifies the key (or the beginning characters of the key) of the last record to be restored. You must enclose the ending value in apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next lower key is the last record restored. For example, if you specify TO-'34', the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record restored.

The TO value must be greater than or equal to the FROM value. If you do not specify the TO parameter, \$COPY uses the last key in the index as the TO key. If only one record is to be restored, the FROM and TO parameters must specify the same key. You cannot specify the TO parameter unless you specify the FROM parameter.

The KEY statement lets you specify the position and length of the key for indexed files. If you enter either the key position or key length, you must also enter the other.

**POSITION-key position**

This specifies the starting position of the key for the restored file. You must specify the key position if the restored file is to be an indexed file, but the saved file was not organized as an indexed file. The key position can be any number from 1 through 4096. If you do not specify a value, and both the saved and restored files are indexed files, the key position for the indexed file that was saved is assumed.

**LENGTH-key length**

This specifies the length of the key for the restored file. You must specify the key length if the restored file is to be an indexed file, but the saved file was not organized as an indexed file. The key length can be any number from 1 through 120. If you do not specify a value, and both the saved and restored files are indexed files, the key length for the saved indexed file is assumed.

**Example 1**

The following example restores all the files that were saved on diskette using the SAVE ALL procedure:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-I1,LABEL-#SAVE
// FILE NAME-COPYO,UNIT-F1
// RUN
// COPYALL TO-F1
// END
```

## \$COPY (RESTORE)

### Example 2

The following example restores a diskette file named FILE1. The key changes from positions 1 through 4 to positions 5 through 24, and only those records that contain the phrase NEW anywhere in the record are copied.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-I1,LABEL-FILE1,LOCATION-S1,AUTO-NO
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE1
// RUN
// COPYFILE OUTPUT-I,INCLUDE-EQ,CHAR-'NEW'
// KEY POSITION-5,LENGTH-20
// END
```

### Example 3

The following example restores all the files that were saved on tape, using the SAVE ALL procedure. The tape is on tape drive 1 and rewinds after the files are restored.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-T1,LABEL-#SAVE,RECFM-FB,RECL-256,
//      BLKL-24576,END-REWIND
// FILE NAME-COPYO,UNIT-F1
// RUN
// COPYALL TO-F1
// END
```

### Example 4

The following example restores a file named DATAFILE from tape cartridge to disk. The disk file name is SAVEDATA.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-TC,LABEL-DATAFILE
// FILE NAME-COPYO,UNIT-F1,LABEL-SAVEDATA
// RUN
// COPYFILE
// END
```

## Saving Files (SAVE Procedure)

See “SAVE Procedure” on page 4-220 for more information.

The following example saves a file on diskette, tape, or tape cartridge:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-file name
// FILE NAME-COPYO,UNIT- $\left\{ \begin{array}{l} I1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ ,LABEL-file name

// RUN
// COPYFILE  $\left[ \begin{array}{l} \text{OUTPUT-} \left\{ \begin{array}{l} \text{SAME} \\ \text{SEQUENTL} \\ (S) \\ \text{INDEXED} \\ (I) \\ \text{DIRECT} \\ (D) \end{array} \right\} \right] \left[ \begin{array}{l} \text{,REORG-} \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \left[ \begin{array}{l} \text{,INCLUDE-} \left\{ \begin{array}{l} \text{EQ} \\ \text{NE} \\ \text{LT} \\ \text{LE} \\ \text{GT} \\ \text{GE} \end{array} \right\} \right] \\ \\ \left[ \text{,POSITION-position} \right] \left[ \text{,CHAR-} \left\{ \begin{array}{l} \text{'characters'} \\ \text{Xddd...dd} \end{array} \right\} \right] \\ \\ \left[ \text{,RECL-record length} \right] \left[ \text{,LIMIT-record limit} \right] \left[ \text{,COMPRESS-} \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \\ \\ \left[ \begin{array}{l} \text{// SELECT } \left\{ \begin{array}{l} \text{RECORD} \\ \text{KEY} \\ \text{PKY} \end{array} \right\} \text{,FROM-starting value} \left[ \text{,TO-ending value} \right] \right] \\ \\ \left[ \begin{array}{l} \text{// KEY POSITION-key position,LENGTH-key length} \right] \\ \\ \text{// END}$ 
```

59020373-1

The following example saves all files, a specific file group, or all files except file groups on diskette, tape, or tape cartridge:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1
// FILE NAME-COPYO,UNIT- $\left\{ \begin{array}{l} I1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ ,LABEL-set name

// RUN
// COPYALL TO- $\left\{ \begin{array}{l} I1 \\ T1 \\ T2 \\ TC \end{array} \right\}$   $\left[ \begin{array}{l} \text{,GROUP-} \left\{ \begin{array}{l} \text{ALL} \\ \text{file group} \end{array} \right\} \right] \left[ \begin{array}{l} \text{,COMPRESS-} \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \\ \\ \text{// END}$ 
```

59020374-1

The following example adds a disk file to an existing diskette file:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-file name
// FILE NAME-COPYO,UNIT-I1,LABEL-file name,PACK-volume id
// RUN
// COPYADD
// END
```

59020375-0

Not all FILE OCL statement parameters are shown. For more information about the FILE OCL statements, see “FILE OCL Statement (for Disk Files)” on page 5-25, “FILE OCL Statement (for Diskette Files)” on page 5-33, and “FILE OCL Statement (for Tape Files)” on page 5-37.

## Differences from the Procedure Parameters

If you specified T1, T2, or TC for the UNIT parameter in the FILE statement, you must also specify RECFM-FB, RECL-256, and BLKL-24576.

For the COPYFILE statement:

### OUTPUT-SAME

This specifies that the restored file will have the same organization as the file being saved. If you do not specify a parameter, SAME is assumed.

### OUTPUT-SEQUENTL

This specifies that the restored file will be organized as a sequential file.

### OUTPUT-INDEXED

This specifies that the restored file will be organized as an indexed file.

### OUTPUT-DIRECT

This specifies that the restored file will be organized as a direct file.

### CHAR-'characters'

This lets you specify up to 30 characters of data. The data must begin and end with apostrophes ('). You may specify apostrophes in the character string by entering two apostrophes. For example, enter o'clock as 'o'clock'.

If your system recognizes DBCS data, the character string can contain a mixture of DBCS and non-DBCS characters. Enclose double-byte character strings in a shift-out (hex 0E) character and a shift-in (hex 0F) character. If there is a shift-out character in the first position of the total character string, that shift-out character is not included for comparison. Likewise, if there is a shift-in character in the last position of the string, that shift-in character is not included for comparison. All other shift-out and shift-in characters are part of the comparison string. Each DBCS character occupies 2 characters in the character string, while each shift-out and shift-in character occupies 1 character.

**Note:** If string of non-DBCS characters has an EBCDIC value equivalent to the DBCS characters you have specified in the comparison string, and it lies in the specified position, that record containing the string of non-double-byte characters can be included or omitted (depending on what you specified earlier for the INCLUDE or OMIT keyword). Take care when dealing with files whose records are not all in the same format.

### CHAR-Xdddd...dd

This specifies the comparison characters in hexadecimal form. You may specify up to 15 bytes, where 2 digits (dd) represent the hexadecimal byte. The hexadecimal digits are 0 through 9 and A through F. The X must precede the hexadecimal digits.

### RECL

This specifies the record length of the new file, and can be any number from 1 through 4096. If you do not enter this parameter, the record length of the file being saved is used.

If the record length of the file being saved is less than the entered record length, the additional record positions in the saved file are filled with blanks. If the record length of the file being saved is greater than the entered record length, the extra positions are truncated. If the new

file is to be an indexed file and the key field would be truncated, an error message is issued.

**LIMIT** This specifies the total number of records to be copied into the new file. You may enter any number greater than 0.

**Note:** The system cannot create a file that contains more than 16 777 200 records.

The SELECT utility control statement lets you further select which records are to be saved. The parameters for the SELECT statements are as follows:

**RECORD** This specifies that a portion of the file is to be saved. When you specify RECORD, the FROM parameter and the TO parameter (if you use TO) must specify relative record numbers.

**FROM-starting value**

This specifies the relative record number of the first record to be saved. For example, if the first record to be saved is the fifth record in the file, you must specify FROM-5. If only one record is to be saved, the FROM and TO parameters must specify the same relative record number. The FROM value must be less than or equal to the TO value.

**TO-last record**

This specifies the relative record number of the last record to be saved. For example, if the last record to be saved is the fifteenth record in the file, you must specify TO-15. If only one record is to be saved, the FROM and TO parameters must specify the same relative record number. If you do not specify the TO parameter, \$COPY uses the last record in the file as the TO record. The TO value must be greater than or equal to the FROM value. You cannot specify the TO parameter unless you specify the FROM parameter.

**KEY or PKY**

This specifies that a specified portion of an indexed file is to be saved. You must specify PKY if the indexed file contains packed keys. If the file contains packed keys, you may specify up to 239 numeric characters. If the file contains regular keys, you may specify up to 120 characters.

**FROM-starting value**

This specifies the key (or the beginning characters of the key) of the first record to be saved. You must enclose the starting value in apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next higher key is the first record saved. For example, if you specify FROM-'15', the first key beginning with 15 or more is the key of the first record saved.

The FROM value must be less than or equal to the TO value. If only one record is to be saved, the FROM and TO parameters must specify the same key.

**TO-ending value**

This specifies the key (or the beginning characters of the key) of the last record to be saved. You must enclose the ending value in apostrophes ('). If none of the keys in the

file begins with the specified characters, the record with the next lower key is the last record saved. For example, if you specify TO-'34', the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record saved.

The TO value must be greater than or equal to the FROM value. If only one record is to be saved, the FROM and TO parameters must specify the same key. If you do not specify the TO parameter, \$COPY uses the last key in the index as the TO key. You cannot specify the TO parameter unless you specify the FROM parameter.

The KEY statement lets you specify the position and length of the key for a newly created indexed file. Enter both the key position and the key length whenever you want to change either value.

**POSITION-key position**

This specifies the starting position of the key for the saved file. The key position can be any number from 1 through 4096. If you do not specify a value, the current key position for the indexed file being saved is assumed.

**LENGTH-key length**

This specifies the length of the key for the saved file. The key length can be any number from 1 through 120. If you do not specify a value, the current key length for the indexed file being saved is assumed.

For the COPYALL statement:

**GROUP** This specifies whether members of a file group are to be saved. If you do not specify a GROUP parameter, only files that are not members of file groups are saved.

**ALL** This specifies that all files and all members of file groups are to be saved.

**group name**

This specifies that all the members of the specified file group are to be saved. Those files that are not part of the file group are not saved.

For the COPYADD statement:

**COPYADD**

This specifies that a disk file is to be added to an existing diskette file. COPYADD creates extension files following the existing diskette file.

**Note:** The previously saved file and the file that is to be added to the previously saved file must have the same attributes.

**Example 1**

The following example saves all disk files on diskette for a period of seven days. The diskettes have volume IDs of VOL001. The files are compressed.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1
// FILE NAME-COPYO,UNIT-I1,LABEL-#SAVE,PACK-VOL001,RETAIN-7
// RUN
// COPYALL TO-I1,GROUP-ALL,COMPRESS-YES
// END
```

**Example 2**

The following example saves a file named FILE1 and adds this file to an existing diskette file named FILE2. The volume ID of the diskette is VOL001.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-I1,LABEL-FILE2,PACK-VOL001
// RUN
// COPYADD
// END
```

**Example 3**

The following example saves all files belonging to file group A1 to diskette. The name of the set of saved files is SAVEA1. The volume ID of the diskette is VOL002, and the files are saved for 39 days.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1
// FILE NAME-COPYO,UNIT-I1,LABEL-SAVEA1,PACK-VOL002,RETAIN-39
// RUN
// COPYALL TO-I1,GROUP-A1
// END
```

**Example 4**

The following example saves specific records from a single file named FILE3 onto a diskette. The only records saved contain the word SAVE in positions 10 through 13 of the records.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE3
// FILE NAME-COPYO,UNIT-I1,LABEL-FILE3,PACK-VOL001
// RUN
// COPYFILE INCLUDE-EQ,POSITION-10,CHAR-'SAVE'
// END
```

**Example 5**

The following example saves only records 5 through 10 of a disk file named FILE4 to diskette. The volume ID of the diskette is VOL001.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE4
// FILE NAME-COPYO,UNIT-I1,LABEL-FILE4,PACK-VOL001
// RUN
// COPYFILE
// SELECT RECORD,FROM-5,TO-10
// END
```

## \$COPY (System/34 Compatible)

### Example 6

The following example saves a disk file named FILE4 to tape. The volume ID of the tape is VOL001 and it is saved to tape drive 2. After the save, the tape position is left at the end of the file.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-FILE4,UNIT-F1
// FILE NAME-COPYO,LABEL-FILE4,UNIT-T2,RECFM-FB,
// RECL-256,BLKL-24576,VOLID-VOL001,END-LEAVE
// RUN
// COPYFILE
// END
```

### Example 7

The following example saves a disk file named DATAFILE to tape cartridge. The name of the file on tape cartridge is SAVEDATA.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-DATAFILE
// FILE NAME-COPYO,UNIT-TC,LABEL-SAVEDATA,RECL-256,BLKL-24576,RECFM-FB,
// VOLID-TEST1
// RUN
// COPYFILE
// END
```

## Saving the Network Resource Directory (SAVENRD Procedure)

Not supported. See "SAVENRD Procedure" on page 4-230 for more information.

## IBM System/34 Compatible Statements

These statements are supported for compatibility with IBM System/34.

The following example copies a file:

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-input file name,UNIT-{F1}
// FILE NAME-COPYO,LABEL-output file name,UNIT-{F1}
// RUN
// COPYFILE OUTPUT-DISK [ ,DELETE- { 'position,c'
//                               'position,cc'
//                               'position,Xddd'
//                               'position,Xdddd'
//                               'position,DBCS-constant'
//                               SYSDEL } ] [ ,REORG- { NO
//                                                  YES } ]
// SELECT { RECORD
//          KEY
//          PKY } ,FROM-starting value [ ,TO-ending value ]
// KEY POSITION-key position,LENGTH-key length
// END
```

59020664-0

The following example lists a file:

```

// LOAD $COPY
// FILE NAME-COPYIN,LABEL-input file name,UNIT-{F1}
// RUN

// COPYFILE {OUTPUT-PRINT} [ ,DELETE-{'position,c',
//                               'position,cc',
//                               'position,Xdd',
//                               'position,Xdddd',
//                               'position,DBCS-constant'} ] [ ,REORG-{'NO',
//                               'YES'} ]
//                               {OUTPTX-PRINT}

[ // SELECT {RECORD} ,FROM-starting value [ ,TO-ending value ]
  {KEY
  {PKY
// END

```

SF020665-0

Not all FILE OCL statement parameters are shown. For information about the FILE OCL statement, see "FILE OCL Statement (for Diskette Files)" on page 5-33 and "FILE OCL Statement (for Disk Files)" on page 5-25.

The following describes how the System/36 environment utility program differs from the System/36 environment procedure.

For the COPYFILE statement:

#### **OUTPUT-DISK**

This specifies that the output file is to be located on disk or diskette.

The COPYIN and COPYO FILE OCL statements indicate the file to be copied (COPYIN) and the new file (COPYO) to be created.

#### **OUTPUT-PRINT or OUTPTX-PRINT**

This specifies listing of all or part of a file on the system list device for the display station.

**DELETE** This specifies that certain records are to be excluded from being copied or listed, based upon their contents. The DELETE parameter is optional.

#### **'position,c'**

This specifies exclusion of any record that has the specified character in the specified position. For example, DELETE-'50,S' excludes all records that have an S in position 50.

#### **'position,cc'**

This specifies exclusion of any record that has the 2 specified characters in the specified beginning position. For example, DELETE-'50,ST' deletes all records that have ST in positions 50 and 51.

#### **'position,Xdd'**

This specifies exclusion of any record having the specified character in the specified position. Xdd is the hexadecimal value of the character. For example, DELETE-'50,X31' excludes all records that have hex 31 in position 50.

#### **'position,Xdddd'**

This specifies exclusion of any record having the 2 specified characters in the specified beginning position. Xdddd are the hexadecimal values of the 2 characters. For example, DELETE-'50,X3132' excludes all records having hex 3132 in positions 50 and 51.

**'position,DBCS-constant'**

This specifies exclusion of any record having the specified DBCS character in the specified position. You must bracket the DBCS constant portion of the parameter with the shift-out (hex 0E) and shift-in (hex 0F) characters.

**Warning:** If 2 non-DBCS characters are in the specified position and happen to have an EBCDIC value equivalent to the 2 bytes of the DBCS character, that record is excluded. Take care when dealing with files whose records are not all in the same format.

**SYSDEL** This specifies exclusion of any system-deleted records.

**REORG-YES**

This specifies the following:

- For an indexed file, records are to be copied or listed sequentially by key. Also, deleted records are not to be copied.
- For sequential files, deleted records are not to be copied.

**REORG-NO**

This specifies that records are to be copied or listed in the order they appear in the file. Deleted records are to be copied. If you do not specify a REORG parameter, REORG-NO is assumed.

The SELECT utility control statement lets you further select which records are to be copied or listed.

**RECORD** This specifies copying of a portion of the file. When you specify RECORD, the FROM parameter and the TO parameter (if you use TO) must specify relative record numbers.

**FROM-starting value**

This specifies the relative record number of the first record to be copied. For example, if the first record to be copied is the fifth record in the file, you must specify FROM-5. If only one record is to be copied, the FROM and TO parameters must specify the same relative record number. The FROM value must be less than or equal to the TO value.

**TO-last record**

This specifies the relative number of the last record to be copied. For example, if the last record to be copied is the fifteenth record in the file, you must specify TO-15. If only one record is to be copied, the FROM and TO parameters must specify the same relative record numbers. If you do not specify the TO parameter, \$COPY uses the last record in the file as the TO record.

The TO value must be greater than or equal to the FROM value. You cannot specify the TO parameter without specifying the FROM parameter.

**KEY or PKY**

This specifies copying of a specified portion of an indexed file. You must specify PKY if the indexed file contains packed keys. If the file contains packed keys, you may specify up to 239 numeric characters. If the file contains regular keys, you may specify up to 120 characters.

**FROM-starting value**

This specifies the key (or the beginning characters of the key) of the first record to be copied. You must enclose the starting value in apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next higher key is the first record to be copied. For example, if you specify FROM-'15', the first key beginning with 15 or larger is the key of the first record to be copied.

The FROM value must be less than or equal to the TO value. If only one record is to be copied, the FROM and TO parameters must specify the same key.

**TO-ending value**

This specifies the key (or the beginning characters of the key) of the last record to be copied. You must enclose the ending value in apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next lower key is the last record copied. For example, if you specify TO-'34', the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record copied.

The TO value must be greater than or equal to the FROM value. If only one record is to be copied, the FROM and TO parameters must specify the same key. If you do not specify the TO parameter, \$COPY uses the last key in the index as the TO key.

You cannot specify the TO parameter without specifying the FROM parameter.

The KEY statement lets you specify the position and length of the key for indexed files. Enter both the key position and the key length whenever you want to change either value.

**POSITION-key position**

This specifies the starting position of the key for a new indexed file. You must specify the key position if the new file is to be indexed and the old file was not an indexed file. Specify the key position as any number from 1 through 4096. If you do not specify a value, and both the old and new files are indexed, the system assumes the key position of the old indexed file.

**LENGTH-key length**

This specifies the length of the key for a new indexed file. You must specify the key length if the new file is to be indexed, but the old file was not organized as an indexed file. Specify the key length as any number from 1 through 120. If you do not specify a value, and both the old and new files are indexed, the system assumes the key length of the old indexed file.

## \$CZUT (ALERT)

### Example 1

The following example saves a disk file named FILEA on diskette:

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-FILEA,UNIT-F1
// FILE NAME-COPYO,LABEL-FILEA,PACK-VOL001,UNIT-I1
// RUN
// COPYFILE OUTPUT-DISK
// END
```

### Example 2

The following example lists a disk file named FILEA:

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-FILEA,UNIT-F1
// RUN
// COPYFILE OUTPUT-PRINT
// END
```

---

## \$CPPE Utility

The \$CPPE utility program lets you display a specified error message (ERR procedure).

See "ERR Procedure" on page 4-93 for more information.

```
// LOAD $CPPE
// RUN
// ERR MIC-message id code,CONTROL-options,ALPHA-USER
// END
```

59020378-0

## Differences from the Procedure Parameters

### ALPHA-USER

This specifies usage of the current level-one-user-message member to retrieve the message.

### Example

The following example uses the \$CPPE utility to display a message with a message identification code (MIC) number of 0014. The allowed options are 0, 1, and 3.

```
// LOAD $CPPE
// RUN
// ERR MIC-0014,CONTROL-013,ALPHA-USER
// END
```

---

## \$CZUT Utility

Not supported. See "ALERT Procedure" on page 4-4 for more information.

---

## \$DCOPY Utility

Not supported. See the “COPYDIAG Procedure” on page 4-55 for more information.

---

## \$DDST Utility

The \$DDST utility program is supported as a no-operation function. Only syntax checking of utility control statements is done.

```

// LOAD $DDST
// RUN

// KEYSORT LABEL-file name [ ,DATE- $\left\{ \begin{array}{l} \text{mmddy} \\ \text{ddmmy} \\ \text{yymmdd} \end{array} \right\} \left[ \begin{array}{l} \text{RETAIN-} \left\{ \begin{array}{l} \text{J} \\ \text{T} \end{array} \right\} \right] \\ \\ \left[ \begin{array}{l} \text{CHECKDUP-} \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \\ \\ // END$ 
```

59020380-0

---

## \$DELET Utility

The \$DELET utility program lets you remove files, libraries, or folders from disk or diskette.

See “DELETE Procedure” on page 4-76 for more information.

The following example deletes one or more files from diskette:

```

// LOAD $DELET
// RUN

// { SCRATCH } LABEL- $\left\{ \begin{array}{l} \text{file name} \\ \text{ALL} \end{array} \right\}$ , UNIT-I1 [ ,DATE- $\left\{ \begin{array}{l} \text{mmddy} \\ \text{ddmmy} \\ \text{yymmdd} \end{array} \right\} \left[ \begin{array}{l} \text{LOCATION-} \left\{ \begin{array}{l} \text{S1} \\ \text{S2} \\ \text{S3} \\ \text{M1.nn} \\ \text{M2.nn} \end{array} \right\} \right] \\ \\ \left[ \begin{array}{l} \text{ENDLOC-} \left\{ \begin{array}{l} \text{S1} \\ \text{S2} \\ \text{S3} \\ \text{M1.nn} \\ \text{M2.nn} \end{array} \right\} \right] \left[ \text{PACK-volume id} \right] \left[ \text{DATA-} \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \\ \\ // END$ 
```

59020381-0

The following example deletes a file, library, folder, or a group of files from disk:

## \$DELET (DELETE)

```
// LOAD $DELET
// RUN

// { SCRATCH } LABEL- { file name
                       library name
                       folder name
                       ALL
                       } ,UNIT-F1 [ ,DATE- { mddy
                                           ddmyy
                                           yymmdd
                                           } ] [ ,USERLIBS- { NO
                                                         YES
                                                         } ]

                       [ ,FOLDER- { NO
                                   YES
                                   } ] [ ,GROUP- { file group
                                                ALL
                                                } ] [ ,DATA- { NO
                                                            YES
                                                            } ]

// END
```

59020382-0

You can repeat the SCRATCH and REMOVE utility control statements to delete several files, libraries, or folders with only one load and run of the \$DELET utility program.

### Differences from the Procedure Parameters

**DATA** This specifies whether to erase the data in the file, library, or folder. The DATA parameter is allowed only if you use the REMOVE utility control statement. DATA-YES does the same function as the ERASE parameter in the DELETE procedure.

### USERLIBS

This specifies whether to delete one or more libraries. YES specifies deletion of libraries. This parameter also deletes files if you specify FOLDER-YES or LABEL-ALL. NO specifies no deletion of libraries. You must be authorized to a library and the objects it contains to delete it. You cannot delete a library if one or more objects in the library are in use. All objects in the library that can be deleted are deleted.

**FOLDER** This specifies whether to delete one or more folders. YES specifies deletion of folders. This parameter also deletes files if you specify USERLIBS-YES or LABEL-ALL. NO specifies no deletion of folders. To delete the folder, you must be authorized to all documents or folders in a document folder, and the documents and folders must not be in use. You must be enrolled in the system distribution directory.

### GROUP-ALL

This specifies deletion of all files on disk if you specify UNIT-F1 and LABEL-ALL. If you specify LABEL-ALL, UNIT-F1, USERLIBS-YES, FOLDER-YES, and GROUP-ALL, all files, libraries, and folders on disk are deleted. If you specify LABEL-ALL and UNIT-F1, and do not specify the GROUP keyword, all files not part of a file group are deleted.

If password security is active, only an operator who has a security classification of security officer can specify ALL. If password security is not active, you can only specify ALL at the system console.

You must have all object authority to do LABEL-ALL or GROUP-ALL deletes.

**Note:** You must specify USERLIBS-YES and FOLDER-YES to prevent the file from being deleted if a file and library or folder have the same name.

**Example 1**

The following example deletes the file named PAYROLL from the diskette in slot S1:

```
// LOAD $DELET
// RUN
// REMOVE LABEL-PAYROLL,UNIT-I1
// END
```

**Example 2**

The following example deletes three files and two libraries with one load and run of the \$DELET utility program:

```
// LOAD $DELET
// RUN
// REMOVE LABEL-FILE1,UNIT-F1
// REMOVE LABEL-FILE2,UNIT-F1
// REMOVE LABEL-FILE3,UNIT-F1
// REMOVE LABEL-LIBRARY1,UNIT-F1,USERLIBS-YES
// REMOVE LABEL-LIBRARY2,UNIT-F1,USERLIBS-YES
// END
```

**Example 3**

The following example deletes all files, libraries, and folders from disk. This example does not delete system libraries and files. However, it does delete any program product libraries.

```
// LOAD $DELET
// RUN
// REMOVE LABEL-ALL,UNIT-F1,USERLIBS-YES,FOLDER-YES,GROUP-ALL
// END
```

---

**\$DPGP Utility**

The \$DPGP utility program lets you print a graphics file on an intelligent printer data stream (IPDS) printer.

See "PRTGRAPH Procedure" on page 4-186 for more information.

```
// LOAD $DPGP
// LOCAL OFFSET-1,DATA-'',AREA-SYSTEM
// LOCAL OFFSET-1,DATA-'file name',AREA-SYSTEM
// LOCAL OFFSET-10,DATA-'width',AREA-SYSTEM
// RUN
```

S9020566-0

**Example**

The following example prints a graphics file named FILE1, with a width of 13, on an IPDS printer:

```
// LOAD $DPGP
// LOCAL OFFSET-1,DATA-'',AREA-SYSTEM
// LOCAL OFFSET-1,DATA-'FILE1',AREA-SYSTEM
// LOCAL OFFSET-10,DATA-'13',AREA-SYSTEM
// RUN
```

---

## \$DPGR Utility

The \$DPGR utility programs allow you to build a graphics file that contains the graphic options and parameters for printing with other data print files.

See "BLDGRAPH Procedure" on page 4-19 for more information.

```
// LOAD $DPGR
// LOCAL OFFSET-1,DATA-'          ',AREA-SYSTEM
// LOCAL OFFSET-1,DATA-'member name',AREA-SYSTEM
// LOCAL OFFSET-9,DATA-'library name',AREA-SYSTEM
// LOCAL OFFSET-19,DATA-'file name',AREA-SYSTEM
// RUN
```

RV2P180-0

### Example

The following example builds a graphics file with a name of GRAPHFL1 from the data in source member GRAPH1 which is in library MYLIB:

```
// LOAD $DPGR
// LOCAL OFFSET-1,DATA-'          ',AREA-SYSTEM
// LOCAL OFFSET-1,DATA-'GRAPH1',AREA-SYSTEM
// LOCAL OFFSET-9,DATA-'MYLIB',AREA-SYSTEM
// LOCAL OFFSET-19,DATA-'GRAPHFL1',AREA-SYSTEM
// RUN
```

---

## \$DUPRD Utility

The \$DUPRD utility program lets you copy all files or libraries on a diskette to another diskette. The System/36 environment does not support copying a single file.

See "COPY11 Procedure" on page 4-55 for more information.

```
// LOAD $DUPRD
// FILE NAME-COPY11,UNIT-I1 [ ,DATE- { mmdyy
                                { ddmyy
                                { yymmdd } ] ]

// RUN

// COPY11 NAME- { ALL } ,PACK-volume id [ ,DELETE- { NO
                                           { YES } } [ ,PRESERVE- { NO
                                                         { YES } }

[ ,COPIES- { 1 } ] [ ,LOCATION- { input slot location } ]

[ ,TOLOC-output slot location ] [ ,CHECK- { NO
                                           { YES } } ]

// END
```

59020383-1

**Example**

The following example copies files on a diskette to another diskette:

```
// LOAD $DUPRD
// FILE NAME-COPYI1,UNIT-I1
// RUN
// COPYI1 NAME-ALL
// END
```

**\$FBLD Utility**

The \$FBLD utility program lets you do the following:

- Create a new, empty disk file (BLDFILE procedure).
- Create an alternative index for a physical file (BLDINDEX procedure).

**Creating Files (BLDFILE Procedure)**

See "BLDFILE Procedure" on page 4-17 for more information.

```
// LOAD $FBLD
// RUN
// FILE LABEL-file name,ATTRIB- $\left\{ \begin{array}{l} \text{SEQUENTL} \\ \text{(S)} \\ \text{INDEXED} \\ \text{(I)} \\ \text{DIRECT} \\ \text{(D)} \end{array} \right\}$ , $\left\{ \begin{array}{l} \text{BLOCKS-size} \\ \text{B-size} \\ \text{RECORDS-size} \\ \text{R-size} \end{array} \right\}$ ,RECL-record length

 $\left[ \begin{array}{l} \text{,LOCATION-} \\ \left\{ \begin{array}{l} \text{A1} \\ \text{A2} \\ \text{A3} \\ \text{A4} \\ \text{block number} \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} \text{,RETAIN-} \\ \left\{ \begin{array}{l} \text{T} \\ \text{S} \\ \text{U} \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} \text{,DFILE-} \\ \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \end{array} \right]$ 

 $\left[ \begin{array}{l} \text{,POSITION-key position,LENGTH-key length} \end{array} \right]$ 

 $\left[ \begin{array}{l} \text{,DUPKEY-} \\ \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} \text{,EXTEND-} \\ \text{extend value} \\ \left\{ \begin{array}{l} 0 \\ 1 \end{array} \right\} \end{array} \right]$ 

// END
```

59020666-1

You can also repeat the FILE utility control statement to build several files with only one load and run of the \$FBLD utility program.

**Example**

The following example creates a resident, delete-capable file with a length of 13 blocks. The file is INVOICE, the record length is 50 bytes, and each record contains a 4-byte key beginning at position 9 in the record.

```
// LOAD $FBLD
// RUN
// FILE LABEL-INVOICE,ATTRIB-I,BLOCKS-13,RECL-50,
//     RETAIN-T,DFILE-YES,
//     POSITION-9,LENGTH-4
// END
```

## \$FREE (COMPRESS)

### Creating Alternative Indexes (BLDINDEX Procedure)

See "BLDINDEX Procedure" on page 4-20 for more information.

```
// LOAD $FBLD
// RUN
// FILE LABEL-alternative index file name,ATTRIB-ALTINDEX,
//                                     {X}

PLABEL-physical file name [ ,PDATE- { mmddyy
                                     ddmmyy
                                     yyymmdd } ] { POSITION-key position,
                                     POSITION1-key position }

{ LENGTH-key length } [ POSITIN2-key position,LENGTH2-key length ]

[ ,POSITIN3-key position,LENGTH3-key length ]

[ ,LOCATION- { A1
              A2
              A3
              A4
              block location } ] [ ,DUPKEY- { YES
                                              NO } ]

// END
```

59020385-1

#### Example

The following example creates an alternative index file from a physical indexed file named CUSTOMER. The CUSTOMER file is assigned a customer number in positions 1 through 4 of the record. The alternative index file is identified by the customer's name in positions 5 through 20 of the record (a key length of 16). The alternative index file is named CUSTNAME.

```
// LOAD $FBLD
// RUN
// FILE LABEL-CUSTNAME,ATTRIB-ALTINDEX,PLABEL-CUSTOMER,
//     POSITION-5,LENGTH-16,DUPKEY-YES
// END
```

---

## \$FREE Utility

The \$FREE utility program is supported as a no-operation function. Only syntax checking of the utility control statements is done.

```
// LOAD $FREE
// RUN

// COMPRESS [ DISK- { ALL
                   A1
                   A2
                   A3
                   A4 } ] [ ,FREE- { LOW
                                   HIGH } ]

// END
```

59020386-0

---

**\$HELP Utility**

See “HELP Procedure” on page 4-109 for more information. You cannot run \$HELP by entering LOAD and RUN OCL statements from the keyboard.

---

**\$HIST Utility**

Not supported. See the “HISTORY Procedure” on page 4-115 for more information.

---

**\$IDSET Utility**

Not supported. See the “DEFINEID Procedure” on page 4-73 for more information.

---

**\$IEDS Utility**

Not supported. See the “DISABLE Procedure” on page 4-81 for more information.

---

**\$IENBL Utility**

Not supported. See the “ENABLED (Enabled Communications) Condition” on page 3-32 for more information.

---

**\$INIT Utility**

The \$INIT utility program lets you initialize a diskette.

See “INIT Procedure” on page 4-119 for more information.

```

// LOAD $INIT
// RUN

// UIN OPTION- { RENAME
                { FORMAT
                { FORMAT2
                { DELETE
                { DIAG
                }
, LOCATION- { S1
             { S2
             { S3
             { M1
             { M2
             { M1.nn
             { M2.nn
             }
, ENDDLOC- { S1
            { S2
            { S3
            { M1
            { M2
            { M1.nn
            { M2.nn
            }

, ACTFLMSG- { YES
            { NO
            }
, RECL- { record length
        { 80
        }

// VOL PACK-volume id,ID-owner id
// END

```

59020391-1

**Differences from the Procedure Parameters****ACTFLMSG**

This specifies whether to check for active files. The INIT procedure always checks for active files. YES specifies that \$INIT checks for active files. If any active files exist on the diskette, a message appears. NO specifies that \$INIT does not check for active files.



**Example 1**

The following example displays the contents of a diskette in slot S1:

```
// LOAD $LABEL
// RUN
// DISPLAY LABEL-ALL,UNIT-I1
// END
```

**Example 2**

The following example displays DDM files:

```
// LOAD $LABEL
// RUN
// DISPLAY LABEL-ALL,UNIT-F1,LOCATION-#NRD.FLE
// END
```

**Example 3**

The following example lists the names of all disk files, libraries, and folders to a disk file called OUTFILE:

```
// LOAD $LABEL
// RUN
// DISPLAY LABEL-ALL,UNIT-F1,OUTPUT-OUTFILE
// END
```

---

**\$MAINT Utility**

The \$MAINT utility program lets you do the following:

- Create a library on disk (BLDLIBR procedure).
- Create source or procedure members.
- Change information about a library member (CHNGEMEM procedure).
- Copy a member from one library to another (LIBRLIBR procedure).
- Copy a member from a library to disk, diskette, tape, or tape cartridge (FROMLIBR procedure).
- Copy members from disk, diskette, tape, or tape cartridge to a library (TOLIBR procedure).
- Start a job from disk, diskette, tape, or tape cartridge (JOBSTR procedure).
- List library members and information (LISTLIBR procedure).
- List information about libraries on disk, diskette, tape, or tape cartridge (LISTFILE procedure).
- Remove members from a library (REMOVE procedure).
- Save a library on diskette, tape, or tape cartridge (SAVELIBR procedure).
- Restore a library from diskette, tape, or tape cartridge (RESTLIBR procedure).

Also shown are \$MAINT utility control statements supported for compatibility with IBM System/34, and the format of the COPY and CEND statements used by the \$MAINT utility program.

**Create a Library (BLDLIBR Procedure)**

See "BLDLIBR Procedure" on page 4-22 for more information.

## \$MAINT (BLDLIBR)

```
// LOAD $MAINT

// FILE NAME-file name,UNIT- $\left\{ \begin{array}{l} I1 \\ F1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ 

// RUN
// ALLOCATE LIBRNAME-library name,LIBRSIZE-library size,STATUS-CREATE

        [ ,DIRSIZE-directory size ] [ ,LOCATION- $\left\{ \begin{array}{l} A1 \\ A2 \\ A3 \\ A4 \\ \text{block number} \end{array} \right\}$  ]

// COPY FROM- $\left\{ \begin{array}{l} \text{DISK} \\ \text{TAPE} \end{array} \right\}$ ,TO-library name,FILE-file name

// END
```

59020393-1

Not all of the FILE OCL statement parameters are shown. For more information about the FILE statement, see “FILE OCL Statement (for Diskette Files)” on page 5-33, “FILE OCL Statement (for Disk Files)” on page 5-25, or “FILE OCL Statement (for Tape Files)” on page 5-37.

### Differences from the Procedure Parameters STATUS-CREATE

This specifies creation of the library.

LIBRSIZE and DIRSIZE are not required and are supported for System/36 compatibility only. The values specified for these two keywords are syntax checked, but are not used when the library is created.

If the file containing members to copy is a record-mode file, each member in the file must begin with a COPY statement and end with a CEND statement. For the formats of the COPY and CEND statements, see “COPY and CEND Statements” on page A-47. The COPY and CEND statements are automatically inserted into members created by \$MAINT. Insert them at the beginning and end of members that are not created by \$MAINT. Do not insert more than the one required CEND statement, however. If a CEND statement exists within the member, an error message appears.

If the record-mode file is organized as a direct file, insert an END statement following the CEND statement that ends the last member in the file. The format of the END statement is:

```
// END
```

where only one blank must separate the // and the END.

### Example

The following example creates a new library called MYLIB:

```
// LOAD $MAINT
// RUN
// ALLOCATE STATUS-CREATE,LIBRNAME-MYLIB,LIBRSIZE-100,DIRSIZE-10
// END
```

## Create Source or Procedure Members

The \$MAINT utility program lets you create source and procedure members.

The easiest way to create library source or procedure members is to use a program such as SEU, which lets you enter and change library source and procedure members.

If you do not have SEU, use the \$MAINT utility program to create source or procedure members entered from the keyboard.

**Note:** You can only create members using \$MAINT. You cannot change a statement in a library member using \$MAINT.

The format of the statements to create a source or procedure member is:

```

// LOAD $MAINT
// RUN
// COPY FROM-READER,LIBRARY- $\left\{ \begin{array}{l} P \\ S \end{array} \right\}$ ,NAME-member name,TO- $\left\{ \begin{array}{l} \text{library name} \\ F1 \end{array} \right\}$ 

       $\left[ \begin{array}{l} ,RETAIN- $\left\{ \begin{array}{l} P \\ R \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} ,RECL- $\left\{ \begin{array}{l} \text{statement length} \\ 120 \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} ,MRT-YES \\ \end{array} \right] \left[ \begin{array}{l} ,PDATA- $\left\{ \begin{array}{l} NO \\ YES \end{array} \right\} \end{array} \right]$ 

       $\left[ \begin{array}{l} ,HIST- $\left\{ \begin{array}{l} YES \\ NO \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} ,REF-number \\ \end{array} \right] \left[ \begin{array}{l} ,SUB-subtype \\ \end{array} \right]$ 

statements for library member

// CEND
// END$$$$ 
```

59020394-0

### FROM-READER

This specifies that the information for the \$MAINT program is supplied from SYSIN. Enter the OCL statements from the keyboard or include them in a procedure.

### LIBRARY-P

This specifies placement of the statements being entered into a library procedure member. Procedure members are members in a source file named QS36PRC in the specified library.

### LIBRARY-S

This specifies placement of the statements being entered into a library source member. Source members are the members in a source file named QS36SRC in the specified library.

### NAME-member name

This specifies the name assigned to the library member. A member name does not exceed 8 characters and begins with an alphabetic character (A through Z, #, \$, or @). Do not use periods (.) within a member name. The remaining characters are any combination of characters (numeric, alphabetic, and special). Avoid using the following characters, because they have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater-than signs (>), plus signs (+), minus signs (-), and equal signs (=). Do not use DIR, LIBRARY, or ALL as a member name.

## \$MAINT (BLDLIBR)

### **TO-library name**

This specifies the name of the library to contain the member. F1 specifies the system library (#LIBRARY).

### **RETAIN-P**

This specifies that if a library member with the same name as **member name** exists in the library, a message appears. The operator can select an option to replace the existing member or cancel the \$MAINT program. If you do not specify the RETAIN parameter, the system assumes RETAIN-P.

### **RETAIN-R**

This specifies replacement of the existing member, if a library member with the same name as the new member name exists in the library.

### **RECL-statement length**

This specifies the length of the statements, in characters, for the source or procedure member, and is any decimal number from 40 to 120. If you do not specify the RECL parameter, 120 is assumed.

**MRT-YES** This specifies the procedure member as a multiple requester terminal (MRT) procedure. If you do not specify MRT-YES, the member is not a MRT procedure.

### **PDATA-YES**

This specifies the passing of data (not parameters) on the procedure command when the procedure starts. The data starts with the first non-blank character following the procedure name and ends with the last nonblank character in the statement. The data is passed on the first input operation from the requester display station in the first program of the procedure. Every MRT procedure has this attribute whether or not you specify PDATA.

Data passed on the INCLUDE OCL statement can contain DBCS characters. However, parameters passed on the INCLUDE OCL statement cannot contain DBCS characters.

### **PDATA-NO**

This specifies passing of parameters to the procedure if the procedure is not a MRT procedure.

### **HIST-YES**

This specifies logging of the OCL statements contained in the procedure to the job log. If you do not specify the HIST parameter, HIST-YES is assumed.

**HIST-NO** This specifies that the OCL statements contained in the procedure are not logged to the job log.

### **REF-number**

This specifies the reference number for the member. Specify up to 6 digits. For example, 000042 is the forty-second revision of a member. For more information, see "Reference Numbers for Library Members" on page 1-18.

### **SUB-subtype**

This specifies the subtype for the members to be created. If not specified, the subtype UNS is used. The following table shows valid subtypes.

Subtype to Enter	Subtype Displayed	Meaning
ARP	RPT or RPT36	RPG auto report member
ARS	ARS36	Automatic response member
ASM	ASM36	Assembler member
BAP	BASP or BASP36	BASIC procedure (source member)
BAS	BAS or BAS36	BASIC member
BGC	SBGC36	Business graphics chart
BGD	BGD36	Business graphics data
BGF	BGF36	Business graphics format
COB	CBL or CBL36	COBOL member
CSM		Communications and system management member (see following note)
CSP		Cross-system product member (see following note)
DFU	DFU36	Data file utility member
DTA	DTA36	Data member
FMT	DSPF or DSPF36	Display format member
FOR	FOR36	FORTRAN member
ICF		CONFIGIC procedure member (see following note)
KEY		KEYS procedure member (see following note)
MNU	MNU or MNU36	Menu member
MSG	MSGF or MSGF36	Message member
PHL	PHL36	Phone list member
QDE		Query data entry member (see following note)
QRY		Query member (see following note)
RPG	RPG or RPG36	RPG member
SRT	SRT36	Sort member
SSP		CNFIGSSP procedure member (see following note)
TXT	TXT	Text member
UNS	UNS36	Unspecified
WSU	WSU36	Work station utility member
X25		X.25 packet switching control link (see following note)

**Note:** This subtype is supported for compatibility only. OS/400 subtypes do not correspond to these System/36 subtypes. If you specify this subtype, no members are found.

## \$MAINT (ALOCLIBR)

### statements for library member

This specifies the statements for the library member being created.  
Enter at least one statement between the COPY and CEND statements.

### Example

The following example copies the following procedure into a procedure member named PROC1. The name of the library to contain the procedure is MYLIB. The statements in the procedure are as follows:

```
// LOAD PROG1
// FILE NAME-FILE1
// RUN
```

The procedure runs a program named PROG1. The procedure is not a MRT procedure.

The following statements are needed to create the procedure:

```
// LOAD $MAINT
// RUN
// COPY FROM-READER,LIBRARY-P,NAME-PROC1,TO-MYLIB,RETAIN-R
// LOAD PROG1
// FILE NAME-FILE1
// RUN
// CEND
// END
```

## Change Library or Directory Size (ALOCLIBR Procedure)

The ALOCLIBR procedure is supported as a no-operation command. The system does only a syntax check of the parameters.

```
// LOAD $MAINT
// RUN

// ALLOCATE LIBRNAME-library name,STATUS- $\left\{ \begin{array}{l} \text{REALOC} \\ \text{INCR} \\ \text{DECR} \end{array} \right\} \left[ \text{,LIBRSIZE-library size} \right]$ 

 $\left[ \text{,DIRSIZE-directory size} \right] \left[ \text{,LOCATION-} \left\{ \begin{array}{l} \text{A1} \\ \text{A2} \\ \text{A3} \\ \text{A4} \\ \text{block number} \end{array} \right\} \right]$ 

// END
```

SP020395-0

## Change Library Member Information (CHNGEMEM Procedure)

See "CHNGEMEM Procedure" on page 4-40 for more information.

```
// LOAD $MAINT
// RUN

// CHANGE NAME- {member name
                {member name.ALL}
                ALL} , LIBRARY- {S
                                P
                                O
                                R
                                ALL} [ , LIBRNAME- {library name}
                                         {#LIBRARY} ]

        [ ,NEWNAME-new member name ] [ ,SUB-subtype ]

        [ ,REF-reference number ]

// END
```

59020397-0

### Example

The following example changes the name of the library procedure member PAYROLL1 to PAYROLL2 in the library named FINANCE:

```
// LOAD $MAINT
// RUN
// CHANGE NAME-PAYROLL1, LIBRARY-P, LIBRNAME-FINANCE,
NEWNAME-PAYROLL2
// END
```

## Gather Unused Library Space (CONDENSE Procedure)

The CONDENSE procedure is supported as a no-operation command. The system does only a syntax check of the parameters.

```
// LOAD $MAINT
// RUN

// COMPRESS [ LIBRNAME- {library name}
             {#LIBRARY} ]

// END
```

59020398-0

## Copy Members from One Library to Another (LIBRLIBR Procedure)

See "LIBRLIBR Procedure" on page 4-129 for more information.

```
// LOAD $MAINT
// RUN

// COPY FROM- $\left\{ \begin{array}{l} \text{from library name} \\ \text{F1} \end{array} \right\}$ ,LIBRARY- $\left\{ \begin{array}{l} \text{S} \\ \text{P} \\ \text{O} \\ \text{R} \\ \text{ALL} \end{array} \right\}$ ,NAME- $\left\{ \begin{array}{l} \text{member name} \\ \text{member name.ALL} \\ \text{ALL} \end{array} \right\}$ ,

TO- $\left\{ \begin{array}{l} \text{to library name} \\ \text{F1} \end{array} \right\}$ ,NEWNAME-new member name  $\left[ \text{,RETAIN-} \left\{ \begin{array}{l} \text{P} \\ \text{R} \end{array} \right\} \right]$ 

 $\left[ \text{,OMIT-} \left\{ \begin{array}{l} \text{member name} \\ \text{member name.ALL} \\ \text{SYSTEM} \end{array} \right\} \right]$ 

// END
```

59020399-0

### Differences from the Procedure Parameters

**OMIT** This specifies the name of one or more members omitted from the copy:

**member name**

This specifies omitting of members with the specified name.

**member name,ALL**

This specifies omitting of all members whose names begin with the specified characters. Specify up to 7 characters.

**SYSTEM** This is ignored in the System/36 environment.

### Example 1

The following example copies all the library members named TEST from a library named MYLIB into a library named YOURLIB, and replaces any members named TEST in YOURLIB:

```
// LOAD $MAINT
// RUN
// COPY FROM-MYLIB,TO-YOURLIB,NAME-TEST,LIBRARY-ALL,
// RETAIN-R
// END
```

### Example 2

The following example copies all the procedures beginning with PAY from a library named MYLIB to a library named YOURLIB. However, those library members beginning with PAYR do not copy.

```
// LOAD $MAINT
// RUN
// COPY FROM-MYLIB,TO-YOURLIB,NAME-PAY.ALL,LIBRARY-P,
// RETAIN-R,OMIT-PAYR.ALL
// END
```

## Copy Members from a Library (FROMLIBR Procedure)

See "FROMLIBR Procedure" on page 4-103 for more information.

These statements allow you to do more tasks than the FROMLIBR procedure allows. You can, for example:

- Create a basic data exchange diskette file that contains one or more library members.
- Omit one or more members from the copy operation.

### To Create a Sector-Mode File

To create sector-mode files, specify the TO-DISK or TO-TAPE parameter without the RECL parameter. You can specify a sector-mode copy for any type of library member (source, procedure, load, or subroutine). In sector mode, copies are in the system format and consist of control information followed by the member as it exists in the library. The file created is an AS/400 save file when using sector mode.

```
// LOAD $MAINT
// FILE NAME- $\left\{ \begin{array}{l} \text{file name} \\ \text{member name} \end{array} \right\}$ ,UNIT- $\left\{ \begin{array}{l} \text{I1} \\ \text{F1} \\ \text{T1} \\ \text{T2} \\ \text{TC} \end{array} \right\}$ ,PACK-volume id
                                ,VOLID-volume id
                                ,VOLID-volume id

// RUN
// COPY FROM- $\left\{ \begin{array}{l} \text{library name} \\ \text{F1} \end{array} \right\}$ ,TO- $\left\{ \begin{array}{l} \text{DISK} \\ \text{TAPE} \end{array} \right\}$ ,FILE- $\left\{ \begin{array}{l} \text{file name} \\ \text{member name} \end{array} \right\}$  ,

NAME- $\left\{ \begin{array}{l} \text{member name} \\ \text{member name.ALL} \\ \text{ALL} \end{array} \right\}$ ,LIBRARY- $\left\{ \begin{array}{l} \text{S} \\ \text{P} \\ \text{R} \\ \text{O} \\ \text{ALL} \end{array} \right\}$  ,ADD- $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$ 

 $\left[ \begin{array}{l} \text{,OMIT-} \left\{ \begin{array}{l} \text{SYSTEM} \\ \text{member name} \\ \text{member name.ALL} \end{array} \right\} \end{array} \right]$ 

// END
```

59020667-0

### Differences from the Procedure Parameters

If you specify T1, T2, or TC for the UNIT parameter in the FILE statement, specify RECFM-FB.

**OMIT** This specifies the name of one or more library members omitted from the copy function.

#### member name

This specifies omitting of members with the specified name from the copy function.

#### member name.ALL

This specifies omitting of all members whose names begin with the specified characters from the copy function. Specify up to 7 characters.

**SYSTEM** This is ignored by the System/36 environment.

## To Create a Record-Mode File

The RECL parameter indicates that the copy of a file is in record mode, not sector mode. You can specify record mode only for source and procedure members. Source and procedure member copies made in record mode are preceded by a COPY statement and followed by a CEND statement.

```
// LOAD $MAINT
// FILE NAME- $\left\{ \begin{array}{l} \text{file name} \\ \text{member name} \end{array} \right\}$ ,UNIT- $\left\{ \begin{array}{l} \text{I1} \\ \text{F1} \\ \text{T1} \\ \text{T2} \\ \text{TC} \end{array} \right\}$ ,PACK-volume id
//                                $\left\{ \begin{array}{l} \text{VOLID-volume id} \\ \text{VOLID-volume id} \end{array} \right\}$ 
// RUN
// COPY FROM- $\left\{ \begin{array}{l} \text{library name} \\ \text{F1} \end{array} \right\}$ ,TO- $\left\{ \begin{array}{l} \text{DISK} \\ \text{TAPE} \end{array} \right\}$ ,FILE- $\left\{ \begin{array}{l} \text{file name} \\ \text{member name} \end{array} \right\}$ ,
//
// NAME- $\left\{ \begin{array}{l} \text{member name} \\ \text{member name.ALL} \\ \text{ALL} \end{array} \right\}$ ,LIBRARY- $\left\{ \begin{array}{l} \text{S} \\ \text{P} \\ \text{ALL} \end{array} \right\}$ , $\left\{ \begin{array}{l} \text{RECL-record length} \\ \text{ADD-YES} \end{array} \right\}$ 
//
//  $\left[ \begin{array}{l} \text{,OMIT-} \\ \left\{ \begin{array}{l} \text{SYSTEM} \\ \text{member name} \\ \text{member name.ALL} \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} \text{,BASIC-} \\ \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \end{array} \right] \left[ \begin{array}{l} \text{,SVATTR-} \\ \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \end{array} \right]$ 
// END
```

59020401-1

For more information about the FILE statement, see “FILE OCL Statement (for Disk Files)” on page 5-25, “FILE OCL Statement (for Diskette Files)” on page 5-33, and “FILE OCL Statement (for Tape Files)” on page 5-37.

## Differences from the Procedure Parameters

If you specify T1, T2, or TC for the UNIT parameter in the FILE statement, specify RECFM-FB.

**TO-DISK** This specifies copying of the library members to a disk or diskette file.

**TO-TAPE** This specifies copying of the library members to a tape file.

### LIBRARY-ALL

When a sector-mode file is created, this specifies copying of all source (S), procedure (P), subroutine (R), and load (O) members; and certain AS/400 object types (see “FROMLIBR Procedure” on page 4-103).

When a record-mode file is created, ALL specifies copying of all S and P members.

### ADD

This specifies the addition of the selected source and procedure members to a record-mode disk file. The disk file must have been created either by a previous record-mode copy, or by using the BLDFILE procedure to create a resident sequential file with a record length between 40 and 120.

When ADD-YES is specified, the RECL parameter must also be specified to indicate that the copy is to be done in record-mode. The record length of the existing file will be used instead of the RECL value.

If ADD-YES is specified when copying members to diskette or tape, or when copying members in sector-mode to a disk file, an error message appears.

**RECL** This specifies the record length, in bytes, of a source or procedure member. The record length can be from 40 through 120. Specify a record length of at least 73 bytes if you also specify SVATTR=YES, or an error message appears.

**BASIC** This specifies whether to place the copied output in a basic data exchange diskette file.

**NO** This specifies that a basic data exchange diskette file is not created. If you do not specify BASIC, the system assumes BASIC=NO.

**YES** This specifies that a basic data exchange diskette file is created.

### Example 1

The following example saves a library procedure member named TEST on a diskette with a volume ID of VOL003. The procedure is in the library named MYLIB.

```
// LOAD $MAINT
// FILE NAME-TEST,UNIT-I1,PACK-VOL003
// RUN
// COPY FROM-MYLIB,TO-DISK,LIBRARY-P,NAME-TEST,FILE-TEST
// END
```

### Example 2

The following example creates a record-mode file from a library source member. The name of the file is FILE1 and it has a record length of 80. The library member is named TEST and is in the library named MYLIB.

```
// LOAD $MAINT
// FILE NAME-FILE1,UNIT-F1,BLOCKS-30
// RUN
// COPY FROM-MYLIB,TO-DISK,LIBRARY-S,NAME-TEST,FILE-FILE1,
// RECL-80
// END
```

### Example 3

The following example creates a basic data exchange diskette record-mode file from a library source member. The name of the file is FILE2 and it has a record length of 80. The library member is named TEST and is in the library named MYLIB. The diskette used has a volume ID of VOL001.

```
// LOAD $MAINT
// FILE NAME-FILE2,UNIT-I1,PACK-VOL001
// RUN
// COPY FROM-MYLIB,TO-DISK,LIBRARY-S,NAME-TEST,FILE-FILE2,
// RECL-80,BASIC=YES
// END
```

## COPY and CEND Statements

When copying one or more library members to a record-mode file, the \$MAINT utility adds a COPY statement before each member and a CEND statement after each member.

The COPY statement has the following format:

```
// COPY NAME-member name,LIBRARY- $\left\{ \begin{matrix} S \\ P \end{matrix} \right\}$  [ ,MRT-YES ] [ ,PDATA-YES ] [ ,HIST-NO ]

[ ,DATE- $\left\{ \begin{matrix} mm/dd/yy \\ dd/mm/yy \\ yy/mm/dd \end{matrix} \right\}$  ] [ ,TIME-hhmm ] [ ,REF-nnnnnn ] [ ,SUB-subtype ]
```

59020409-2

**NAME-member name**

This specifies the library member placed in the specified library.

**LIBRARY-S**

This specifies the member as a source member.

**LIBRARY-P**

This specifies the member as a procedure member.

**MRT-YES** This specifies the procedure member as a MRT procedure. If you do not specify MRT-YES, the member is not a MRT procedure.

**PDATA-YES**

This specifies that data (not parameters) can pass on the procedure command that causes the procedure to run. The data starts with the first nonblank character following the procedure name and ends with the last nonblank character in the statement. The data is passed on the first input operation from the requester display station, from the first program in the procedure. Every MRT procedure has this indicator, whether or not you specify PDATA-YES.

If you do not specify PDATA-YES and if the procedure is not a MRT procedure, parameters are passed to the procedure.

**HIST-NO** This specifies that the OCL statements in the procedure should not be logged to the job log. If you do not specify HIST-NO, the OCL statements in the procedure are logged to the job log.

**DATE** This specifies the date the member was created or last changed. You must specify the date in the same format as the session date.

**TIME** This specifies the time that the member was created or last changed. The time is in the form: hhmm (hours, minutes).

**REF** This specifies the reference number of the member. The number is 6 digits long and is right-justified (for example, 000042 is the forty-second revision of a member). For further information, see "Reference Numbers for Library Members" on page 1-18.

**SUB** This specifies the subtype for the members to be processed. If not specified, the subtype is not used as a qualifier when selecting a members. The following table shows valid subtypes:

Subtype to Enter	Subtype Displayed	Meaning
ARP	RPT or RPT36	RPG auto report member
ARS	ARS36	Automatic response member
ASM	ASM36	Assembler member
BAP	BASP or BASP36	BASIC procedure (source member)
BAS	BAS or BAS36	BASIC member

<b>Subtype to Enter</b>	<b>Subtype Displayed</b>	<b>Meaning</b>
BGC	BGC36	Business graphics chart
BGD	BGD36	Business graphics data
BGF	BGF36	Business graphics format
COB	CBL or CBL36	COBOL member
CSM		Communications and system management member (see following note)
CSP		Cross-system product member (see following note)
DFU	DFU36	Data file utility member
DTA	DTA36	Data member
FMT	DSPF or DSPF36	Display format member
FOR	FOR36	FORTRAN member
ICF		CONFIGIC procedure member (see following note)
KEY		KEYS procedure member (see following note)
MNU	MNU or MNU36	Menu member
MSG	MSGF or MSGF36	Message member
PHL	PHL36	Phone list member
QDE		Query data entry member (see following note)
QRY		Query member (see following note)
RPG	RPG or RPG36	RPG member
SRT	SRT36	Sort member
SSP		CNFIGSSP procedure member (see following note)
TXT	TXT	Text member
UNS	UNS36	Unspecified
WSU	WSU36	Work station utility member
X25		X.25 packet switching control link (see following note)

**Note:** This subtype is supported for compatibility only. OS/400 program subtypes do not correspond to these System/36 subtypes. If you specify this subtype, no members are found.

The CEND statement has the following format:

```
// CEND
```

59020410-0

## Copy Members to a Library (TOLIBR Procedure)

See "TOLIBR Procedure" on page 4-283 for more information.

```
// LOAD $MAINT
// FILE NAME-file name,UNIT- $\left\{ \begin{array}{l} P1 \\ I1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ 

// RUN
// COPY TO- $\left\{ \begin{array}{l} \text{library name} \\ P1 \end{array} \right\}$ ,FROM- $\left\{ \begin{array}{l} \text{DISK} \\ \text{TAPE} \end{array} \right\}$ ,FILE-file name  $\left[ \begin{array}{l} \text{,NAME-} \left\{ \begin{array}{l} \text{member name} \\ \text{member name,ALL} \end{array} \right\} \end{array} \right]$ 

 $\left[ \begin{array}{l} \text{,RETAIN-} \left\{ \begin{array}{l} P \\ R \end{array} \right\} \end{array} \right]$   $\left[ \begin{array}{l} \text{,SUB-subtype} \end{array} \right]$   $\left[ \begin{array}{l} \text{,LIBRARY-} \left\{ \begin{array}{l} S \\ P \\ O \\ R \\ \text{ALL} \end{array} \right\} \end{array} \right]$ 

 $\left[ \begin{array}{l} \text{,OMIT-NEW} \end{array} \right]$ 

// END
```

Not all FILE OCL statement parameters are shown. For information about the FILE OCL statement, see "FILE OCL Statement (for Diskette Files)" on page 5-33, "FILE OCL Statement (for Disk Files)" on page 5-25, and "FILE OCL Statement (for Tape Files)" on page 5-37.

### Differences from the Procedure Parameters

For the COPY statement:

#### FROM-DISK

This specifies copying of the library members from a diskette or disk file.

#### FROM-TAPE

This specifies copying of the library members from a tape file.

#### RETAIN-P

This specifies display of a message if a member with the same name already exists in the library. You can then decide whether to replace the duplicate member. If you do not specify a RETAIN parameter, the system assumes P.

#### RETAIN-R

This specifies replacement of the duplicate member without a message.

#### OMIT-NEW

This specifies that new members are not copied.

### Example

The following example copies the members from a diskette file called PSY into a library named MYLIB and replaces any duplicate members:

```
// LOAD $MAINT
// FILE NAME-PAY,UNIT-I1
// RUN
// COPY FROM-DISK,TO-MYLIB,NAME-PAY,FILE-PAY,RETAIN-R
// END
```

## Start a Job (JOBSTR Procedure)

See "JOBSTR Procedure" on page 4-125 for more information.

```

// LOAD $MAINT
// FILE NAME-file name,UNIT- $\left\{ \begin{array}{l} I1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ 
// RUN

// COPY FROM- $\left\{ \begin{array}{l} DISK \\ TAPE \end{array} \right\}$ ,FILE-file name,TO- $\left\{ \begin{array}{l} \text{library name} \\ F1 \end{array} \right\}$ 

// END

 $\left\{ \begin{array}{l} \text{procedure name} \\ \text{procedure name} \end{array} \right\}$   $\left[ \begin{array}{l} \\ \text{library name} \end{array} \right]$ 

// JOBQ  $\left[ \begin{array}{l} \text{jobq prty} \\ \text{jobq prty} \end{array} \right]$   $\left[ \begin{array}{l} \text{library name} \\ \text{library name} \end{array} \right]$   $\left[ \begin{array}{l} \text{procedure name} \\ \text{procedure name} \end{array} \right]$ 

```

59020403-1

### Example

The following example copies the procedure members (PROC1 and PROC2) and the source member (MEMBER1) into the library named MYLIB from a diskette file named JOBS, and then runs procedure PROC1. See the JOBSTR procedure for the format of the file named JOBS.

```

// LOAD $MAINT
// FILE NAME-JOBS,UNIT-I1
// RUN
// COPY FROM-DISK,FILE-JOBS,TO-MYLIB
// END
PROC1,MYLIB

```

## List Library Members and Information (LISTLIBR Procedure)

See "LISTLIBR Procedure" on page 4-147 for more information.

```

// LOAD $MAINT
// RUN
// COPY TO-PRINT,NAME- $\left\{ \begin{array}{l} \text{DIR} \\ \text{member name} \\ \text{member name.ALL} \\ \text{ALL} \end{array} \right\}$ ,LIBRARY- $\left\{ \begin{array}{l} S \\ P \\ R \\ O \\ \text{ALL} \\ \text{SYSTEM} \end{array} \right\}$   $\left[ \begin{array}{l} \\ \text{MEMBERS} \\ \text{DIRINFO} \end{array} \right]$ ,

FROM- $\left\{ \begin{array}{l} \text{library name} \\ F1 \end{array} \right\}$   $\left[ \begin{array}{l} \\ \text{member name} \\ \text{member name.ALL} \\ \text{SYSTEM} \end{array} \right]$ 

 $\left[ \begin{array}{l} \text{LIST-} \\ \text{USER} \\ \text{DETAIL} \end{array} \right]$   $\left[ \begin{array}{l} \text{PAGE-} \\ \text{NO} \\ \text{YES} \end{array} \right]$   $\left[ \begin{array}{l} \text{SUB-} \\ \text{subtype} \end{array} \right]$ 

 $\left[ \begin{array}{l} \text{PRTFILE-} \\ \text{output file name} \end{array} \right]$ 

// END

```

59020404-1

You can repeat the COPY statement to list several members or libraries with only one load and run of the \$MAINT utility program.

## \$MAINT (LISTFILE)

### Differences from the Procedure Parameters

#### TO-PRINT

This specifies sending of the output to the system list device.

#### OMIT

This specifies one or more library members omitted from the copy or display.

#### member name

This specifies omitting of members with the specified name.

#### member name.ALL

This specifies omitting of all members whose names begin with the specified characters. Specify up to 7 characters.

**SYSTEM** This is ignored in the System/36 environment.

### Example 1

The following example lists all library procedure members that have names beginning with PAY in the library named PAYLIB. However, it does not list the members beginning with PAYR.

```
// LOAD $MAINT
// RUN
// COPY FROM-PAYLIB,TO-PRINT,NAME-PAY.ALL,LIBRARY-P,
//      OMIT-PAYR.ALL
// END
```

### Example 2

The following example lists all the system library procedure members:

```
// LOAD $MAINT
// RUN
// COPY FROM-#LIBRARY,TO-PRINT,NAME-ALL,LIBRARY-P
// END
```

## List Information about Libraries (LISTFILE Procedure)

See "LISTFILE Procedure" on page 4-142 for more information.

```
// LOAD $MAINT
// FILE NAME-file name,UNIT- $\left\{ \begin{array}{l} I1 \\ F1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ 
// RUN
// COPY TO-PRINT,FROM- $\left\{ \begin{array}{l} DISK \\ TAPE \end{array} \right\}$ ,FILE-file name  $\left[ ,LIST- $\left\{ \begin{array}{l} USER \\ DETAIL \end{array} \right\} \right]$ 
// END$ 
```

59020405-1

Not all FILE OCL statement parameters are shown. For information about the FILE OCL statement, see "FILE OCL Statement (for Diskette Files)" on page 5-33, "FILE OCL Statement (for Disk Files)" on page 5-25, and "FILE OCL Statement (for Tape Files)" on page 5-37.

**Differences from the Procedure Parameters**

If you specify T1, T2, or TC for the UNIT parameter in the FILE statement, then you must specify RECFM-FB. If the file listed from T1, T2, or TC is a SAVELIBR file, then you must specify RECL-256, BLKL-24576. If the file listed from T1, T2, or TC is a LIBRFILE, then you must specify RECL-256, BLKL-4096.

For the COPY statement:

**TO-PRINT**

This specifies the sending of output to the system list device.

**FROM-DISK**

This specifies a listing from disk or diskette. A FILE OCL statement must define the file.

**FROM-TAPE**

This specifies a listing from tape. A FILE OCL statement must define the file.

**Remove Members from a Library (REMOVE Procedure)**

See "REMOVE Procedure" on page 4-192 for more information.

```

// LOAD $MAINT
// RUN

// DELETE NAME- {member name
                {member name.ALL}
                {ALL}
                }, LIBRARY- {S
                           P
                           O
                           R
                           ALL}
                        {, LIBRNAME- {library name}
                          {#LIBRARY}}

                [ , OMIT- {member name
                          {member name.ALL} } ] [ , RETAIN- S ]

// END

```

SP020406-0

You can repeat the DELETE utility control statement up to 32 times to remove several library members with only one load and run of the \$MAINT utility program.

**Differences from the Procedure Parameters****LIBRNAME-#LIBRARY**

This specifies removal of members from the system library.

**OMIT-member name**

This specifies that the members having the specified name are not removed from the library. The member name can be from 1 through 8 characters.

**OMIT-member name.ALL**

This specifies that the members beginning with the specified characters are not removed from the library. The member name can be from 1 through 7 characters.

**RETAIN-S**

This is ignored in the System/36 environment.

## \$MAINT (SAVELIBR)

### Example

The following example removes a source member named THIS, a procedure member named THAT, and a load member named OTHER from the library MYLIB:

```
// LOAD $MAINT
// RUN
// DELETE NAME-THIS,LIBRARY-S,LIBRNAME-MYLIB
// DELETE NAME-THAT,LIBRARY-P,LIBRNAME-MYLIB
// DELETE NAME-OTHER,LIBRARY-O,LIBRNAME-MYLIB
// END
```

## Save a Library (SAVELIBR Procedure)

See “SAVELIBR Procedure” on page 4-228 for more information.

```
// LOAD $MAINT

// FILE NAME-library name,UNIT- $\left\{ \begin{array}{l} I1 \\ T1 \\ T2 \\ TC \end{array} \right.$ ,PACK-volume id
                                ,VOLID-volume id
                                ,VOLID-volume id
                                ,VOLID-volume id

// RUN
// COPYLIBR FROM-library name,TO- $\left\{ \begin{array}{l} DISK \\ TAPE \end{array} \right.$ ,FILE-library name

// END
```

59020669-0

Not all FILE OCL statement parameters are shown. For more information about the FILE OCL statement, see “FILE OCL Statement (for Diskette Files)” on page 5-33 and “FILE OCL Statement (for Tape Files)” on page 5-37.

### Example

The following example saves a library named MYLIB on a diskette with a volume ID of VOL001:

```
// LOAD $MAINT
// FILE NAME-MYLIB,UNIT-I1,PACK-VOL001
// RUN
// COPYLIBR FROM-MYLIB,TO-DISK,FILE-MYLIB
// END
```

## Restore a Library (RESTLIBR Procedure)

See "RESTLIBR Procedure" on page 4-203 for more information.

```

// LOAD $MAINT

// FILE NAME-library name,UNIT- $\left\{ \begin{array}{l} I1 \\ T1 \\ T2 \\ TC \end{array} \right\}$ 

// RUN
// COPYLIBR FROM- $\left\{ \begin{array}{l} DISK \\ TAPE \end{array} \right\}$ ,TO-library name,FILE-library name

            $\left[ \begin{array}{l} ,LIBRSIZE-library \text{ size} \\ ,DIRSIZE-directory \text{ size} \end{array} \right]$ 

            $\left[ \begin{array}{l} ,LOCATION- $\left\{ \begin{array}{l} A1 \\ A2 \\ A3 \\ A4 \\ \text{block number} \end{array} \right\}$  \end{array} \right]$ 

// END

```

59020670-0

For more information about the FILE statements, see "FILE OCL Statement (for Diskette Files)" on page 5-33 and "FILE OCL Statement (for Tape Files)" on page 5-37.

### Example

The following example restores a library named MYLIB:

```

// LOAD $MAINT
// FILE NAME-MYLIB,UNIT-I1
// RUN
// COPYLIBR FROM-DISK,TO-MYLIB,FILE-MYLIB
// END

```

---

## \$MGBLD Utility

The \$MGBLD utility program lets you create message load members from message source members.

See "CREATE Procedure" on page 4-66 for more information.

```

// LOAD $MGBLD
// RUN

// MGBLD SOURCE-source member name  $\left[ \begin{array}{l} ,REPLACE- $\left\{ \begin{array}{l} NO \\ YES \end{array} \right\}$  \end{array} \right]$ 

            $\left[ \begin{array}{l} ,LIBRARY- $\left\{ \begin{array}{l} \text{library name} \\ #LIBRARY \end{array} \right\}$  \\ ,HALT- $\left\{ \begin{array}{l} YES \\ NO \end{array} \right\}$  \\ ,SSP- $\left\{ \begin{array}{l} NO \\ YES \end{array} \right\}$  \end{array} \right]$ 

// END

```

59020411-0

## \$PACK (COMPRESS)

### Differences from the Procedure Parameters

- SSP** This specifies whether the load member is protected from deletion.
- YES** This specifies that the authority of the member is changed to \*USE during the creation process and that the load member cannot be removed by the REMOVE procedure. If the creation authority of the library is less than \*USE, the authorities not removed are listed in the job log.
- NO** This specifies that the load member is created with the authority (CRTAUT) of the library it is created into. The operations you can perform on it are determined by the authorities of the object.

If you do not specify the SSP parameter, the system assumes NO.

**Note:** The source member must be in the same library as the message load member being created.

### Example

The following example creates a message load member from a source member named MESSAGES in the library MYLIB:

```
// LOAD $MGBLD
// RUN
// MGBLD SOURCE-MESSAGES,LIBRARY-MYLIB
// END
```

---

## \$MMSP Utility

Not supported. See "STOPM Procedure" on page 4-251 for more information.

---

## \$MMST Utility

Not supported. See "STARTM Procedure" on page 4-249 for more information.

---

## \$PACK Utility

The \$PACK utility program is supported as a no-operation function.

```
// LOAD $PACK
// RUN
```

59020414-0

---

## \$PNLM Utility

Not supported. See "DEFINEPN Procedure" on page 4-73 for more information.

---

## \$POST Utility

Not supported. See "POST Procedure" on page 4-178 for more information.

---

## \$PRCED Utility

Not supported. See "SECEDIT Procedure" on page 4-232 for more information.

---

## \$PRCLT Utility

Not supported. See "SECLIST Procedure" on page 4-233 for more information.

---

## \$PRLST Utility

Not supported. See "SECLIST Procedure" on page 4-233 for more information.

---

## \$PRPWD Utility

The \$PRPWD utility program lets you change your password.

See "PASSWORD Procedure" on page 4-174 for more information.

```
// LOAD $PRPWD  
// RUN
```

59020514-0

### Example

The following example changes your password:

```
// LOAD $PRPWD  
// RUN
```

---

## \$PRUED Utility

Not supported. See "SECEDIT Procedure" on page 4-232 for more information.

---

## \$PRUID Utility

Not supported. See "SECDEF Procedure" on page 4-232 for more information.

---

## \$PRURS Utility

Not supported. See "SECRET Procedure" on page 4-234 for more information.

## \$RRLST (SECLIST)

---

## \$PRUSV Utility

Not supported. See “SECSAVE Procedure” on page 4-234 for more information.

---

## \$RENAM Utility

The \$RENAM utility program lets you rename a file, library, or folder.

See “RENAME Procedure” on page 4-194 for more information.

```
// LOAD $RENAM
// RUN

// RENAME LABEL-current name,NEWLABEL-new name [ ,DATE- { mddy }
                                                    { ddmy }
                                                    { yyddd } ]

// END
```

S9020423-0

You can repeat the RENAME statement to rename several files, libraries, or folders with only one load and run of the \$RENAM utility program.

You can rename data files in your System/36 environment files library, and any libraries or folders to which you are authorized.

### Example

The following example changes the name of an existing file from OLDPAY to NEWPAY, changes THIS to THAT, and changes MYLIB to YOURLIB:

```
// LOAD $RENAM
// RUN
// RENAME LABEL-OLDPAY,NEWLABEL-NEWPAY
// RENAME LABEL-THIS,NEWLABEL-THAT
// RENAME LABEL-MYLIB,NEWLABEL-YOURLIB
// END
```

---

## \$RREDT Utility

Not supported. See “SECEDIT Procedure” on page 4-232 for more information.

---

## \$RRESC Utility

Not supported. See “SECDEF Procedure” on page 4-232 for more information.

---

## \$RRLST Utility

Not supported. See “SECLIST Procedure” on page 4-233 for more information.

---

**\$RRSAV Utility**

Not supported. See “SECSAVE Procedure” on page 4-234 for more information.

---

**\$RRSTR Utility**

Not supported. See “SECRET Procedure” on page 4-234 for more information.

---

**\$RRTED Utility**

Not supported. See “SECEDIT Procedure” on page 4-232 for more information.

---

**\$RRTLTL Utility**

Not supported. See “SECLIST Procedure” on page 4-233 for more information.

---

**\$SETCF Utility**

The \$SETCF utility program lets you establish display station environment items, such as the printer for display station output (SET procedure).

**Change Communications Attributes (ALTERCOM Procedure)**

Not supported. See “ALTERCOM Procedure” on page 4-5 for more information.

**Set Display Station Environment (SET Procedure)**

See “SET Procedure” on page 4-235 for more information.

```

// LOAD $SETCF
[ // IMAGE MEMBER,print belt member name ]
// RUN
// SETCF [ LINES-lines per page ] [ ,IMAGE- $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$  ] [ ,FORMAT- $\left\{ \begin{array}{l} \text{MDY} \\ \text{DMY} \\ \text{YMD} \end{array} \right\}$  ]
[ ,DATE- $\left\{ \begin{array}{l} \text{mddy} \\ \text{ddmmyy} \\ \text{yyymmdd} \end{array} \right\}$  ] [ ,LIBRARY- $\left\{ \begin{array}{l} \text{library name} \\ \# \text{LIBRARY} \\ 0 \end{array} \right\}$  ] [ ,RGSIZE-region size ]
[ ,PRINTER- $\left\{ \begin{array}{l} \text{printer id} \\ \text{SYSTEM} \end{array} \right\}$  ] [ ,FORMSNO-forms number ]
// END

```

59020671-1

**Specify Print Key Information (PRINTKEY Procedure)**

See the “PRINTKEY Procedure” on page 4-182 for more information.

## \$SFGR (FORMAT)

```
// LOAD $SETCF
// RUN

// SETPK [ID-printer id] [ ,BORDER- $\left\{ \begin{array}{l} Y \\ N \end{array} \right\} ] [ ,HEADER- $\left\{ \begin{array}{l} Y \\ N \end{array} \right\} ]$ 

// END$ 
```

RV2P177-0

### Example

The following example shows how to get a border and a header printed.

```
// LOAD $SETCF
// RUN
// SETPK BORDER-Y,HEADER-Y
// END
```

---

## \$SETCP Utility

Not supported. See “SETCOMM Procedure” on page 4-237 for more information.

---

## \$SFGR Utility

The \$SFGR utility program lets you create, update, add, or delete display formats.

See “FORMAT Procedure” on page 4-99 for more information.

For creating, adding to, or updating a display format load member:

```
// LOAD $SFGR
// RUN

// LOADMBR NAME-load member name [ ,REPLACE- $\left\{ \begin{array}{l} NO \\ YES \end{array} \right\} ] [ ,SSP- $\left\{ \begin{array}{l} NO \\ YES \end{array} \right\} ]$ 

// INOUT [ INLIB- $\left\{ \begin{array}{l} \text{source member library} \\ \#LIBRARY \end{array} \right\} ] [ ,OUTLIB- $\left\{ \begin{array}{l} \text{load member library} \\ \#LIBRARY \end{array} \right\} ]$ 

[ ,HALT- $\left\{ \begin{array}{l} NO \\ YES \end{array} \right\} ] [ ,PRINT- $\left\{ \begin{array}{l} YES \\ NO \\ PARTIAL \end{array} \right\} ]$ 

//  $\left\{ \begin{array}{l} CREATE \\ ADD \\ UPDATE \end{array} \right\}$  SOURCE-source member name [ ,NUMBER- $\left\{ \begin{array}{l} \text{number of formats} \\ 1 \end{array} \right\} ]$ 

// END$$$ 
```

59020434-0

### Notes:

1. *Number* must be 1 through 255, but is otherwise ignored.
2. On the AS/400 system, a display load member is an object of type \*FILE. You cannot have more than one object of type \*FILE with the same name in a library. Therefore, you cannot have a display file and other types of files with the same name in the same library.

**Differences from the Procedure Parameters**

You can repeat the ADD, DELETE, and UPDATE statements in any order. Specify up to 32 CREATE statements, or up to 32 ADD, UPDATE, or DELETE statements in any combination. For example, you could specify 16 UPDATE statements, 8 DELETE statements, and 8 ADD statements with only one load and run of the \$\$SFGR utility program. ADD, UPDATE, and DELETE statements are not allowed with a CREATE statement.

**SSP**

This specifies whether the load member is protected from deletion.

**YES** This specifies that the authority of the member is changed to \*USE during the creation process and that the load member cannot be removed by the REMOVE procedure. If the creation authority of the library is less than \*USE, the authorities not removed are listed in the job log.

**NO** This specifies that the load member is created with the authority (CRTAUT) of the library it is created into. The operations you can perform on it are determined by the authorities of the object.

If you do not specify the SSP parameter, the system assumes NO.

**INLIB**

If you do not specify the library, \$\$SFGR uses #LIBRARY. The FORMAT procedure uses the current library as the default library when not deleting.

**OUTLIB**

If you do not specify the library, \$\$SFGR uses #LIBRARY. The FORMAT procedure uses the current library as the default library when not deleting.

The following example deletes a format from a display format load member:

```
// LOAD $$SFGR
// RUN
// LOADMBR NAME-load member name

[ // INOUT OUTLIB-load member library ]

// DELETE FORMAT-display format name
// END
```

59020435-0

You can repeat the DELETE statement. Specify up to 32 statements.

**Example 1**

The following example creates a new display format load member named FORMAT2 in the library named MYLIB. The example uses three display format source members to create the load member. The three source members are named DISP1, DISP2, and DISP3. They are all stored in a library named MYLIB.

```
// LOAD $$SFGR
// RUN
// LOADMBR NAME=FORMAT2,REPLACE=YES
// INOUT INLIB-MYLIB,OUTLIB-MYLIB,PRINT-PARTIAL
// CREATE SOURCE-DISP1,NUMBER-3
// CREATE SOURCE-DISP2,NUMBER-4
// CREATE SOURCE-DISP3,NUMBER-2
// END
```

## \$TCOPY (TAPECOPY)

### Example 2

The following example changes a display format load member named DSPFM by adding new formats, or updating or deleting current formats. The added and updated formats are specified in the corresponding source members.

```
// LOAD $SFGR
// RUN
// LOADMBR NAME-DSPFM
// INOUT INLIB-MYLIB,OUTLIB-MYLIB
// ADD SOURCE-DSP1,NUMBER-3
// UPDATE SOURCE-DSP2,NUMBER-1
// DELETE FORMAT-FMT1
// END
```

---

## \$SINCT Utility

Not supported. See “RESTNRD Procedure” on page 4-205 for more information.

---

## \$SINDL Utility

Not supported. See “DELNRD Procedure” on page 4-80 for more information.

---

## \$SINR Utility

Not supported. See “EDITNRD Procedure” on page 4-86 for more information.

---

## \$SVCASRV Utility

The \$SVCASRV utility is supported as a no-operation function. The system does only a syntax check of the utility control statements.

```
// LOAD $SVCASRV
// RUN

// CACHE [ FUNC- { ALTER }
           { START }
           { STOP } ] [ ,SIZE-size ] [ ,PAGESIZE-pagesize ]

// END
```

59020523-0

---

## \$TCOPY Utility

The \$TCOPY utility program lets you do the following:

- Copy data to or from tape.
- List the contents of exchange files on tape.

**Note:** You cannot use the \$TCOPY utility program with a tape cartridge.

## Copy Data To or From Tape (TAPECOPY Procedure)

See "TAPECOPY Procedure" on page 4-255 for more information.

The following example transfers data from disk to tape or tape to disk:

```

// LOAD $TCOPY
// FILE NAME-COPYIN,UNIT- $\begin{Bmatrix} F1 \\ T1 \\ T2 \end{Bmatrix}$ ,LABEL-from file label

// FILE NAME-COPYO,UNIT- $\begin{Bmatrix} F1 \\ T1 \\ T2 \end{Bmatrix}$ ,LABEL-to file label

// RUN

// TRANSFER  $\begin{bmatrix} \text{ADD-YES} \\ \text{NO} \end{bmatrix}$ , $\begin{bmatrix} \text{CHKLAST-YES} \\ \text{NO} \end{bmatrix}$ , $\begin{bmatrix} \text{KEYLEN-value} \end{bmatrix}$ , $\begin{bmatrix} \text{KEYLOC-value} \end{bmatrix}$ 

// END

```

59020436-0

## List the Contents of Exchange Tape Files (LISTFILE Procedure)

See "LISTFILE Procedure" on page 4-142 for more information.

The following example displays the contents of an exchange tape file:

```

// LOAD $TCOPY
// FILE NAME-COPYIN,UNIT- $\begin{Bmatrix} T1 \\ T2 \end{Bmatrix}$ ,LABEL-from file label

// RUN

// DISPLAY  $\begin{bmatrix} \text{FROM-value} \end{bmatrix}$ , $\begin{bmatrix} \text{TO-value} \end{bmatrix}$ 

// END

```

59020437-1

Not all FILE OCL statement parameters are shown. For more information about the FILE OCL statements, see "FILE OCL Statement (for Disk Files)" on page 5-25 and "FILE OCL Statement (for Tape Files)" on page 5-37.

### Example 1

The following example takes a file on disk named FILE1 and copies it to a tape file named FILE2. The tape is mounted on tape drive 1.

```

// LOAD $TCOPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-T1,LABEL-FILE2
// RUN
// TRANSFER ADD-NO,CHKLAST-YES
// END

```

### Example 2

The following example takes a file from a tape mounted on tape drive 1 named SHOW and displays the first 100 records:



---

## \$UASC Utility

The \$UASC utility program lets you display or print copied spool files from a user file.

See "COPYPRT Procedure" on page 4-57 for more information.

```
// LOAD $UASC
// FILE NAME=file name,DISP-SHR
// RUN
```

59020446-0

### Example

The following example displays the spool file or files copied into a file named SPOOLENT:

```
// LOAD $UASC
// FILE NAME-SPOOLENT
// RUN
```

---

## \$UASF Utility

The \$UASF utility program lets you copy a spool file or files to a disk file.

See "COPYPRT Procedure" on page 4-57 for more information.

```
// LOAD $UASF
// RUN

// SPOOL [ SPOOLID- { ALL
                { spool id }
                { Fxxxx
                { SYSTEM }
            ] [ ,NAME-file name ] [ ,RELCANS- { RELEASE
                { CANCEL } ]

            [ ,RETAIN- { T
                { J
                { S } ]

// END
```

59020447-0

### Differences from the Procedure Parameters

**RETAIN** This classifies the file as a resident (T), job (J), or scratch (S) file when it is copied. If you do not specify a parameter, the system assumes T.

### Example 1

The following example, entered by the system operator, copies all spool files into a file named SPFILE:

```
// LOAD $UASF
// RUN
// SPOOL SPOOLID-SYSTEM,NAME-SPFILE
// END
```

## #GSORT (SORT)

### Example 2

The following example, entered from display station W3, copies all spool files with forms number 0017 into a file named F0017W3:

```
// LOAD $UASF
// RUN
// SPOOL SPOOLID-F0017
// END
```

---

## \$XNLM Utility

Not supported. See "DEFINX21 Procedure" on page 4-74 for more information.

---

## \$XNSH Utility

Not supported. See "DEFINX21 Procedure" on page 4-74 for more information.

---

## \$XREST Utility

Not supported. See "RETEXTN Procedure" on page 4-202 for more information.

---

## \$XSAVE Utility

Not supported. See "SAVEEXTN Procedure" on page 4-227 for more information.

---

## #GCFR Utility

Not supported. See "REQUESTX Procedure" on page 4-195 for more information.

---

## #GSORT Utility

Use the #GSORT utility to sort 1 to 8 input files. The sorting is done by the Reformat Data (FMTDTA) command. #GSORT uses the sort specifications to determine the correct sort sequence.

The sort specifications can be embedded in your procedure, or supplied in a separate source member that is referred to by the // SOURCE statement, as follows:

```
// LOAD #GSORT
// FILE NAME-INPUT,LABEL=label name1
...
[// FILE NAME-WORK,LABEL=label name2,BLOCKS=block size,RETAIN-S]
// FILE NAME-OUTPUT,LABEL=label name3,RETAIN-T,DISP-NEW,RECORDS=record count
// RUN
// SOURCE member name,{#LIBRARY
                      {library name}
--- or ---
insert specifications here in place of the // SOURCE
and follow the sort specifications with an END statement:
// END
```

59020679-1

See "SORT Procedure" on page 4-247 for more information.

The ellipsis (...) after the first // FILE statement indicates that additional // FILE statements can be provided when multiple input files are used requesting a sort/merge.

The //FILE statement for the work file is optional and is ignored when sorting on the AS/400 system, but may be necessary on an actual System/36.

The // END statement should be omitted when the // SOURCE statement is used, but can be supplied within the given source member after the last sort specification.

You can create a procedure that runs a sort job using the following operation control language (OCL) statements:

- LOAD
- FILE
- RUN
- END

You can also use the SOURCE utility control statement. If you intend to sort more than one file, you must use OCL statements because the SORT procedure lets you sort only one file at a time.

The following examples show OCL procedures (created using OCL statements) that can run a sort job.

### **Example 1**

The following OCL procedure calls your sort specifications from a source member:

```
// LOAD #GSORT
// FILE NAME-INPUT,LABEL-ITEMBALN
// FILE NAME-WORK,LABEL-BALANCE,BLOCKS-50,RETAIN-S
// FILE NAME-OUTPUT,LABEL-BALOUT,RETAIN-T,DISP-NEW,RECORDS-300
// RUN
// SOURCE ITEMSORT,ITEMLIB
```

#### **LOAD #GSORT**

This statement tells the system to load an IBM-supplied program called #GSORT into the system. The #GSORT program is the system's sort program.

#### **FILE NAME-INPUT,LABEL-ITEMBALN**

This statement tells the sort program that the name of your input file to be sorted is ITEMBALN.

#### **FILE NAME-WORK,LABEL-BALANCE,BLOCKS-50,RETAIN-S**

This statement tells the sort program that a 50-block area (500 disk sectors) with the label BALANCE is used as the sort work area.

RETAIN-S means that the work file is a scratch file and that it is deleted after the sort is completed.

#### **FILE NAME-OUTPUT,LABEL-BALOUT,RETAIN-T,DISP-NEW, RECORDS-300**

This statement identifies the output file as a new resident file named BALOUT. When this statement is processed, the system reserves disk space for 300 output records.

**RUN** This statement instructs the system to run the #GSORT program.

#### **SOURCE ITEMSORT,ITEMLIB**

This statement identifies to the sort program the name of the source member containing the sort specifications for that sort job. This statement indicates that the sort specifications are contained in a source member named ITEMSORT. The ITEMSORT source member is stored in a library named ITEMLIB.

## #GSORT (SORT)

### Example 2

The following OCL procedure contains OCL statements and sort specifications:

```
// LOAD #GSORT
// FILE NAME-INPUT,LABEL-ITEMBALN
// FILE NAME-WORK,LABEL-BALANCE,BLOCKS-50,RETAIN-S
// FILE NAME-OUTPUT,LABEL-BALOUT,RETAIN-T,DISP-NEW,RECORDS-300
// RUN
    HSORTR      8A      0  24  N
    I P 19 22LEF 23 26          AVAIL FLD LESS OR EQUAL TO REORD
    FNP 19 22                  AVAIL NUMBER OF UNITS AVAILABLE
    FNP 23 26                  REORD REORDER POINT
    FDC  2  9                  ITNBR ITEM NUMBER
// END
```

## File Information Needed When Writing OCL Statements

When you use your own OCL procedure to run a sort job, you must supply the sort program with the information about the input, work, and output files. Use the OCL FILE statement to provide the information to the sort program. For more detailed information about the FILE statement, see "FILE OCL Statement (for Disk Files)" on page 5-25.

### FILE Statement for the Input File

Following is an example FILE statement for an input file:

```
// FILE NAME-INPUT,LABEL-ITEMBALN
```

#### NAME-file name

This statement specifies the name the program uses to refer to the file. For sorting, the file name must be one of the following:

- INPUT or INPUT1 (but not both)
- INPUT2
- INPUT3
- INPUT4
- INPUT5
- INPUT6
- INPUT7
- INPUT8

**Multiple Input Files:** You can use up to eight local and remote files as input to a sort program. You can enter the // FILE statements in any sequence. The input files are processed serially regardless of the order of the // FILE statements. For example, if three // FILE statements were specified for three input files and the file names were ordered INPUT8, INPUT3, and INPUT5, then the order of the sort is INPUT3, INPUT5, and INPUT8. The lowest numbered file is processed first, the next lowest numbered file is processed after that, and so on until all the files have been processed.

Equal control field ordering (H-specification column 12) lets you merge two or more input files. Following are restrictions for multiple input files:

- You must use OCL. The SORT procedure cannot accept multiple input file information.
- SORTA (addrout sort) is not allowed.
- The lengths of the records in each input file must be the same.

- You must label the file:

#### **LABEL-file label**

The label parameter specifies the name by which the input file is identified by you and on the disk.

If you omit the LABEL parameter from a disk FILE statement, the system uses the file name from the NAME parameter. Once you name the input file, you must use the LABEL parameter to give the sort program the label of the input file. For example, use the following format to sort an input file named ITEMBALN:

```
// FILE NAME-INPUT,LABEL-ITEMBALN...
```

#### **FILE Statement for the Work File**

The work file statement is ignored on the AS/400 system. However, it may be required on System/36 in the following instances:

- If the input file is a shared file to which records are added
- If the input file is a shared, open file to which records were added

The work file must be a local file. Following is an example of a FILE statement for a work file:

```
// FILE NAME-WORK,LABEL-ITEMOUT,RECORDS-500
```

The work file is usually a scratch file (RETAIN-S) because you do not need its information after the sort job (or job step) has been run.

#### **FILE Statement for the Output File**

Following is an example of a FILE statement for an output file:

```
// FILE NAME-OUTPUT,LABEL-BALOUT,RECORDS-500,RETAIN-T,DISP-NEW
```

#### **RECORDS or BLOCKS**

This parameter indicates the total number of blocks or records in the output file. You must specify a size large enough to hold all sorted output. If the size you specify is not large enough, an error occurs.

If the output file you want to use is an existing resident (RETAIN-T) file, you must supply its originally-allocated size, or you must indicate that the file exists by specifying DISP-OLD in your output FILE statement. The output file can be remote.

## **#KASRT Utility**

Use the #KASRT utility to sort the following double-byte character sets (DBCS):

- Japanese
- Korean
- Traditional Chinese

Sorting is done by the Reformat Data (FMTDTA) command. #KASRT uses the active sort tables and the information in the sort specifications to determine the correct sort sequence. See the *ADTS/400: Character Generator Utility* book and the *Data Management* book for more information about DBCS sort tables and sorting DBCS characters.

## #KASRT

#KASRT is nearly identical to #GSORT and should only be used on a DBCS version of the operating system. #KASRT supports up to eight input files, and the sort specifications can be embedded in your procedure or supplied in a separate source member that is referred to by the SOURCE statement, as follows:

```
...
// LOAD #KASRT
// FILE NAME-INPUT,LABEL-label name1
...
[// FILE NAME-WORK,LABEL-label name2,BLOCKS-block size,RETAIN-S]
// FILE NAME-OUTPUT,LABEL-label name3,RETAIN-T,DISP-NEW,RECORDS-record count
// RUN
// SOURCE member name,{LIBRARY}
                       {library name}
--- or ---
insert specifications here in place of the // SOURCE
and follow the sort specifications with an END statement:
// END
```

59020680-1

See “SRTX Procedure” on page 4-248 and “#GSORT Utility” on page A-66 for more information.

The // FILE statement for the work file is optional and ignored on the AS/400 system. Place the // END statement at the end of the sort specifications. When you specify the // SOURCE statement, omit the // END statement from the procedure. You must specify the // END statement in the procedure if the // SOURCE statement is replaced in your procedure by embedded sort specifications.

## Appendix B. Converting Decimal and Hexadecimal Numbers

This appendix provides information on converting decimal numbers to hexadecimal numbers, or hexadecimal numbers to decimal numbers. It also provides information on determining the number of blocks a disk file uses when the file is allocated by number of records.

### Converting Decimal to Hexadecimal

Use the following table to convert decimal numbers to hexadecimal numbers, or to convert hexadecimal numbers to decimal numbers.

Byte 2		Position 3		Byte 1		Position 1	
Position 4	Hex Dec	Hex	Dec	Position 2	Hex Dec	Hex	Dec
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	57344	E	3582	E	224	E	14
F	61440	F	3840	F	240	F	15

Examples of how to use this figure follow.

## Hexadecimal to Decimal Example

To find the decimal value of hex 1FA, you would find that:

1. In position 3, hex 1 equals decimal 256.
2. In position 2, hex F equals decimal 240.
3. In position 1, hex A equals decimal 10.

Add these three decimal numbers as follows to get the decimal value of hex 1FA:

$$256 + 240 + 10 = 506$$

Byte 2		Byte 1	
Position 4	Position 3	Position 2	Position 1
Hex Dec	Hex Dec	Hex Dec	Hex Dec
0 0	1 256	F 240	A 10

## Decimal to Hexadecimal Example

To find the hexadecimal value of decimal 538, you would find that:

1. The next lower decimal number in the chart is 512 in position 3. This number is equal to hex 2.
2. Subtract 512 from 538 and use the difference to find the next hexadecimal value as follows:  
 $538 - 512 = 26$
3. The next lower number in the chart from 26 is 16 in position 2. This number is equal to hex 1.
4. Subtract 16 from 26 and use the difference to find the next hexadecimal value as follows:  
 $26 - 16 = 10$
5. The remaining 10 is found in position 1 of the chart. This number is equal to hex A.

When the positions of the hexadecimal values are combined, the resulting hexadecimal value for decimal 538 is hex. Thus, decimal 538 equals hex 21A.

Byte 2		Byte 1	
Position 4	Position 3	Position 2	Position 1
Hex Dec	Hex Dec	Hex Dec	Hex Dec
0 0	2 512	1 160	A 10

---

## Records to Blocks Conversion for Disk Files

This section helps you determine how many blocks a disk file uses when the file is allocated by number of records. One disk block contains 10 sectors. Each sector contains 256 bytes.

### Determining the Number of Blocks in a File

To determine the number of blocks in a file do the following steps:

1. Multiply the number of records by the record length to determine the number of bytes in the data portion of the file as follows:

$$(\text{number of records}) \times (\text{record length}) = (\text{number of bytes})$$

2. Divide the number of bytes by the number of bytes per block. One block contains 2560 bytes as follows:

$$(\text{number of bytes})/2560 = (\text{number of blocks})$$

3. Round the number of blocks up to the next whole number.

#### Example

A sequential disk file was allocated with 200 records. The record length of the file is 15.

1. The number of bytes in the file is 3000:

$$200 \times 15 = 3000$$

2. The number of blocks used by the file is 2:

$$3000/2560 = 1.1 \text{ (round up)}$$

3. Rounding 1.1 up to the next whole number gives 2 as the number of blocks in the file.



## Appendix C. EBCDIC and ASCII Code Tables

The character sets for extended binary-coded decimal interchange code (EBCDIC) and American National Standard Code for Information Interchange (ASCII) are shown in the following tables. Use the set that your system supports.

### EBCDIC

For example, hex C1 (binary 1100 0001) is the letter A.

Main Storage Bit Positions 4,5,6,7		Main Storage Bit Positions 0,1,2,3															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0	NUL	DLE	DS		SP	&	-					{	}	\	0	
0001	1	SOH	DC1	SOS		RSP	/		a	j	~		A	J	NSP	1	
0010	2	STX	DC2	FS	SYN				b	k	s		B	K	S	2	
0011	3	ETX	DC3	WUS	IR				c	l	t		C	L	T	3	
0100	4	SEL	ENP/RES	INP/BYP	PP				d	m	u		D	M	U	4	
0101	5	HT	NL	LF	RS				e	n	v		E	N	V	5	
0110	6		BS	ETB	NBS				f	o	w		F	O	W	6	
0111	7	DEL	POC	ESC	EOT				g	p	x		G	P	X	7	
1000	8	GE	CAN		SBS				h	q	y		H	Q	Y	8	
1001	9	SPS	EM		IT			'	i	r	z		I	R	Z	9	
1010	A	RPT	UBS	SM/SW	RFF	¢	!		:				SHY				
1011	B	VT	CU1	FMT	CU3	.	\$	,	#								
1100	C	FF	IFS			<	•	%	@				⌋		⌈		
1101	D	CR	IGS	ENQ	NAK	(	)	DC4	—	'							
1110	E	SO	IRS	ACK		+	;	>	=				⌋				
1111	F	SI	ITB/IUS	BEL	SUB		⌋	?	"							EQ	

S9020689-0

# ASCII Code Table

## ASCII

For example, hex 41 (binary 0100 0001) is the letter A.

		Main Storage Bit Positions 0,1,2,3																
Main Storage Bit Positions 4,5,6,7	Main Storage Bit Positions		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	Hex		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE	SP	0	Ⓒ	P	·	p									
0001	1	SOH	DC1	!	1	A	Q	a	q									
0010	2	STX	DC2	"	2	B	R	b	r									
0011	3	ETX	DC3	#	3	C	S	c	s									
0100	4	EOT	DC4	\$	4	D	T	d	t									
0101	5	ENQ	NAK	%	5	E	U	e	u									
0110	6	ACK	SYN	&	6	F	V	f	v									
0111	7	BEL	ETB	'	7	G	W	g	w									
1000	8	BS	CAN	(	8	H	X	h	x									
1001	9	HT	EM	)	9	I	Y	i	y									
1010	A	LF	SUB	·	:	J	Z	j	z									
1011	B	VT	ESC	+	;	K	[	k	{									
1100	C	FF	FS	,	<	L	\	l										
1101	D	CR	GS	-	=	M	]	m	}									
1110	E	SO	RS	.	>	N	^	n	~									
1111	F	SI	US	/	?	O	_	o	DEL									

S9020690-0

---

## Appendix D. SFGR System Messages

This appendix lists the printed messages produced by the FORMAT procedure and the Create System/36 Display File (CRTS36DSPF) command. The appendix provides additional information that you can use to correct problems you might have with your display formats. If an error is found when the source specifications are processed, the message is printed immediately following the statement that caused the error.

---

**SYX5050      T VALUE SPECIFIED IN FORM TYPE ENTRY IS INVALID. MUST BE S.**

**Specification Type:** Not applicable

**Explanation:** The first noncomment statement in the display format specifications is not an S-specification (S in column 6).

---

**SYX5051      T INVALID FIRST CHARACTER IN DISPLAY SCREEN FORMAT NAME ENTRY. MUST BE ALPHABETIC, @, # OR \$.**

**Specification Type:** S

**Explanation:** The format name entry is in columns 7 through 14.

---

**SYX5052      T FORMAT NAME ENTRY CONTAINS AN EMBEDDED BLANK.**

**Specification Type:** S

**Explanation:** The format name entry is in columns 7 through 14.

---

**SYX5053      T FORMAT NAME ENTRY CONTAINS A QUOTE OR COMMA.**

**Specification Type:** S

**Explanation:** The format name entry is in columns 7 through 14.

---

**SYX5054      T ADD SPECIFIED, BUT A DUPLICATE DISPLAY SCREEN FORMAT NAME WAS FOUND IN THE FORMAT LOAD MEMBER.**

**Specification Type:** S

**Explanation:** An attempt is being made to add a display format to a format load member, but a format with the name specified in the format name entry (columns 7 through 14) already exists in the format load member.

---

**SYX5055      T UPDATE SPECIFIED, BUT THE FORMAT LOAD MEMBER DOES NOT CONTAIN A FORMAT WITH THE SPECIFIED NAME.**

**Specification Type:** S

**Explanation:** An attempt is being made to update a display format, but a format with the name specified in the format name entry (columns 7 through 14) does not exist in the format load member.

---

**SYX5056      T DUPLICATE FORMAT NAMES IN INPUT SOURCE MEMBERS SPECIFIED FOR THIS RUN OF \$SFGR.**

**Specification Type:** S

**Explanation:** The same format name entry (columns 7 through 14) was specified on two different S-specifications during this run of \$SFGR.

---

**SYX5057      W FIRST POSITION IN START LINE NUMBER ENTRY IS V. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The start line number entry is in columns 17 and 18.

---

**SYX5058      W SECOND POSITION IN START LINE NUMBER ENTRY IS INVALID. 01 IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The start line number entry is in columns 17 and 18.

---

**SYX5059      W VALUE SPECIFIED IN START LINE NUMBER ENTRY IS NOT NUMERIC. 01 IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The start line number entry is in columns 17 and 18.

---

**SYX5060      W VALUE SPECIFIED IN START LINE NUMBER ENTRY EXCEEDS THE NUMBER OF LINES ON THE SCREEN. 01 IS ASSUMED.**

**Specification Type:** S

**Explanation:** The start line number entry is in columns 17 and 18.

---

**SYX5061      W VALUE SPECIFIED IN START LINE NUMBER ENTRY IS ZERO. 01 IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The start line number entry is in columns 17 and 18.

---

**SYX5062      W SECOND POSITION IN NUMBER OF LINES TO CLEAR ENTRY IS NOT NUMERIC. 24 OR 27 IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** For an 80-column screen, 24 is assumed. For a 132-column screen, 27 is assumed. The number of lines to clear entry is in columns 19 and 20.

---

**SYX5063      W VALUE SPECIFIED IN NUMBER OF LINES TO CLEAR ENTRY IS NOT NUMERIC. 24 OR 27 IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** For an 80-column screen, 24 is assumed. For a 132-column screen, 27 is assumed. The number of lines to clear entry is in columns 19 and 20.

---

**SYX5064      W VALUE SPECIFIED IN NUMBER OF LINES TO CLEAR ENTRY EXCEEDS THE NUMBER OF LINES ON THE SCREEN.**

**Specification Type:** S

**Explanation:** The value specified in the number of lines to clear entry (columns 19 and 20) is ignored, and all lines are cleared.

---

**SYX5070      W FIRST POSITION IN SOUND ALARM ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The sound alarm entry is in columns 25 and 26.

---

**SYX5071      W FIRST POSITION IN SOUND ALARM ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The sound alarm entry is in columns 25 and 26.

---

**SYX5072      W SECOND POSITION IN SOUND ALARM ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED.**

**Specification Type:** S

**Explanation:** The sound alarm entry is in columns 25 and 26. Column 25 contains a blank, and column 26 contains a zero or a nonnumeric character.

---

**SYX5073      W INDICATOR SPECIFIED IN SOUND ALARM ENTRY IS NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The sound alarm entry is in columns 25 and 26.

---

**SYX5074      W INDICATOR SPECIFIED IN SOUND ALARM ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The sound alarm entry is in columns 25 and 26.

---

**SYX5075      W ERASE INPUT IS Y AND OVERRIDE FIELDS IS NOT Y. FIELDS ARE IGNORED WHEN THIS FORMAT IS DISPLAYED.**

**Specification Type:** S

**Explanation:** The erase input fields entry is in columns 31 and 32, and the override fields entry is in columns 33 and 34. The D-specifications for the format are checked for errors, but are ignored when the format is displayed.

---

**SYX5076      I KEY MASK ENTRY CONTAINS DUPLICATE NUMBERS OR CHARACTERS. DUPLICATE IS IGNORED.**

**Specification Type:** S

**Explanation:** The key mask entry is in columns 64 through 79.

---

**SYX5077      I FORMAT CONTAINS FIELDS, PART OF A FIELD, OR AN ATTRIBUTE ON A LINE NOT CLEARED BY THE FORMAT.**

**Specification Type:** S or D

**Explanation:** If the fields in this format appear on the display screen along with fields previously displayed, the following rules apply:

- For any output-only field for which no display attributes or indicators are specified in columns 39 through 49 of the D-specification, at least one space must be left between the field and any previously displayed field that follows.
  - For any other field, at least two spaces must be left between the field and any previously displayed field that follows.
- 

**SYX5078      T A PRECEDING INPUT FIELD HAS A POSITION CURSOR ENTRY OF Y.**

**Specification Type:** D

**Explanation:** More than one field has Y specified in the position cursor entry (columns 32 and 33). Y can be specified for only one field in a display.

---

**SYX5080**      **W FIRST POSITION IN BLINK CURSOR ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The blink cursor entry is in columns 29 and 30.

---

**SYX5081**      **W FIRST POSITION IN BLINK CURSOR ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The blink cursor entry is in columns 29 and 30.

---

**SYX5082**      **W SECOND POSITION IN BLINK CURSOR ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The blink cursor entry is in columns 29 and 30. Column 29 contains a blank, and column 30 contains a zero or nonnumeric character.

---

**SYX5083**      **W INDICATOR SPECIFIED IN BLINK CURSOR ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The blink cursor entry is in columns 29 and 30. Valid indicator values are 01 through 99.

---

**SYX5084**      **W INDICATOR SPECIFIED IN BLINK CURSOR ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The blink cursor entry is in columns 29 and 30.

---

**SYX5085**      **W CHARACTER SPECIFIED IN LOWERCASE ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The lowercase entry is in column 21 and must be Y, N, or blank.

---

**SYX5086**      **W FORM TYPE ENTRY IS BLANK. THIS RECORD IS IGNORED.**

**Specification Type:** S, D, or H

**Explanation:** The form type entry is in column 6 and must be either S, D, or H.

---

**SYX5087**      **T FORM TYPE ENTRY IS INVALID. THIS RECORD IS IGNORED.**

**Specification Type:** S, D, or H

**Explanation:** The form type entry is in column 6 and must be either S, D, or H.

---

**SYX5088**      **T FIELD LENGTH ENTRY IS BLANK OR CONTAINS A CHARACTER WHICH IS NOT NUMERIC.**

**Specification Type:** D

**Explanation:** The field length entry is in columns 15 through 18.

---

**SYX5089      T VALUE SPECIFIED IN FIELD LENGTH ENTRY EXCEEDS THE NUMBER OF POSITIONS ON THE SCREEN.**

**Specification Type:** D

**Explanation:** The field length entry is in columns 15 through 18.

---

**SYX5090      T VALUE SPECIFIED IN FIELD LENGTH ENTRY IS ZERO.**

**Specification Type:** D

**Explanation:** The field length entry is in columns 15 through 18.

---

**SYX5091      T SECOND POSITION IN LINE NUMBER ENTRY IS ZERO OR NOT NUMERIC.**

**Specification Type:** D

**Explanation:** The line number entry is in columns 19 and 20. Column 19 contains a blank and column 20 contains a zero or nonnumeric character.

---

**SYX5092      T VALUE SPECIFIED IN LINE NUMBER ENTRY IS NOT NUMERIC.**

**Specification Type:** D

**Explanation:** The line number entry is in columns 19 and 20.

---

**SYX5093      T VALUE SPECIFIED IN LINE NUMBER ENTRY IS ZERO.**

**Specification Type:** D

**Explanation:** The line number entry is in columns 19 and 20.

---

**SYX5094      T VALUE SPECIFIED IN LINE NUMBER ENTRY EXCEEDS THE NUMBER OF LINES ON THE SCREEN.**

**Specification Type:** D

**Explanation:** The line number entry is in columns 19 and 20.

---

**SYX5095      T SECOND POSITION IN HORIZONTAL POSITION ENTRY IS ZERO OR NOT NUMERIC.**

**Specification Type:** D

**Explanation:** The horizontal position entry is in columns 21 and 22. Column 21 contains a blank, and column 22 contains a zero or nonnumeric character.

---

**SYX5096      T VALUE SPECIFIED IN HORIZONTAL POSITION ENTRY IS NOT NUMERIC.**

**Specification Type:** D

**Explanation:** The horizontal position entry is in columns 21 and 22.

---

**SYX5097      T VALUE SPECIFIED IN HORIZONTAL POSITION ENTRY IS ZERO.**

**Specification Type:** D

**Explanation:** The horizontal position entry is in columns 21 and 22.

---

**SYX5098**      **T VALUE IN HORIZONTAL POSITION ENTRY EXCEEDS THE NUMBER OF HORIZONTAL POSITIONS ON THE SCREEN.**

**Specification Type:** D

**Explanation:** The horizontal position entry is in columns 21 and 22.

---

**SYX5099**      **T LINE NUMBER PLUS START LINE NUMBER EXCEEDS THE NUMBER OF LINES ON THE SCREEN.**

**Specification Type:** S and D

**Explanation:** The line number entry (columns 19 and 20 on the D-specification) plus the start line number entry (columns 17 and 18 on the S-specification) exceeds the number of lines on the screen.

---

**SYX5100**      **W SCREEN POSITION IS EQUAL TO 0101. IF VARIABLE START LINE NUMBER IS SET TO 1, AN ERROR WILL OCCUR.**

**Specification Type:** S and D

**Explanation:** On the D-specification, the line number entry (columns 19 and 20) and the horizontal position entry (columns 21 and 22 ) were both 01; and, on the S-specification, a variable line number was specified (V in column 17). Because a field cannot begin in position 0101, an error will occur if the variable start line number is 01 when the format is displayed.

---

**SYX5101**      **T SCREEN POSITION SPECIFIED IS EQUAL TO 0101. THIS POSITION CANNOT BE USED.**

**Specification Type:** S and D

**Explanation:** On the D-specification, the line number entry (columns 19 and 20) and the horizontal position entry (columns 21 and 22) were both 01; and, on the S-specification, the start line number entry (columns 17 and 18) was 01. However, a field cannot begin in position 0101.

---

**SYX5102**      **T THIS FIELD OCCUPIES A SCREEN POSITION ALREADY DEFINED BY A PREVIOUS FIELD IN THIS FORMAT.**

**Specification Type:** D

**Explanation:** This field occupies one or more positions of a field previously defined in this format. Fields are not allowed to overlap.

---

**SYX5103**      **T MORE THAN 256 FIELDS WERE DEFINED FOR THIS FORMAT.**

**Specification Type:** D

**Explanation:** A maximum of 256 fields is allowed for a display.

---

**SYX5104**      **W FIRST POSITION IN OUTPUT DATA ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The output data entry is in columns 23 and 24.

---

**SYX5105**      **W FIRST POSITION IN OUTPUT DATA ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The output data entry is in columns 23 and 24.

---

**SYX5106 W SECOND POSITION IN OUTPUT DATA ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The output data entry is in columns 23 and 24. Column 23 contains a blank, and column 24 contains a zero or nonnumeric character.

---

**SYX5107 W INDICATOR SPECIFIED IN OUTPUT DATA ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The output data entry is in columns 23 and 24. Valid indicator values are 01 through 99.

---

**SYX5108 W INDICATOR SPECIFIED IN OUTPUT DATA ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The output data entry is in columns 23 and 24.

---

**SYX5109 W CHARACTER SPECIFIED IN INPUT ALLOWED ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The input allowed entry is in column 26 and must be Y, N, or blank.

---

**SYX5110 W FIRST POSITION IN PROTECT FIELD ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The protect field entry is in columns 37 and 38.

---

**SYX5111 W FIRST POSITION IN PROTECT FIELD ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The protect field entry is in columns 37 and 38.

---

**SYX5112 W SECOND POSITION IN PROTECT FIELD ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The protect field entry is in columns 37 and 38. Column 37 contains a blank, and column 38 contains a zero or nonnumeric character.

---

**SYX5113 W INDICATOR SPECIFIED IN PROTECT FIELD ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The protect field entry is in columns 37 and 38. Valid indicator values are 01 through 99.

---

**SYX5114**      **W INDICATOR SPECIFIED IN PROTECT FIELD ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The protect field entry is in columns 37 and 38.

---

**SYX5115**      **W FIRST POSITION IN POSITION CURSOR ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The position cursor entry is in columns 32 and 33.

---

**SYX5116**      **W FIRST POSITION IN POSITION CURSOR ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The position cursor entry is in columns 32 and 33.

---

**SYX5117**      **W SECOND POSITION IN POSITION CURSOR ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The position cursor entry is in columns 32 and 33. Column 32 contains a blank, and column 33 contains a zero or nonnumeric character.

---

**SYX5118**      **W INDICATOR SPECIFIED IN POSITION CURSOR ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The position cursor entry is in columns 32 and 33. Valid indicator values are 01 through 99.

---

**SYX5119**      **W INDICATOR SPECIFIED IN POSITION CURSOR ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The position cursor entry is in columns 32 and 33.

---

**SYX5120**      **W CHARACTER SPECIFIED IN MANDATORY ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The mandatory entry is in column 29 and must be Y, N, or blank.

---

**SYX5121**      **W CHARACTER SPECIFIED IN MANDATORY FILL ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The mandatory fill entry is in column 28 and must be Y, N, or blank.

---

**SYX5122**      **W CHARACTER SPECIFIED IN ADJUST/FILL ENTRY IS INVALID. BLANK IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The adjust/fill entry is in column 31 and must be Z, B, or blank.

---

---

**SYX5123      W CHARACTER SPECIFIED IN COLUMN SEPARATORS ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The column separators entry is in column 49 and must be Y, N, or blank.

---

**SYX5124      W CHARACTER SPECIFIED IN CONTROLLED FIELD EXIT ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The controlled field exit entry is in column 35 and must be Y, N, or blank.

---

**SYX5125      W CHARACTER SPECIFIED IN AUTO RECORD ADVANCE ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The auto record advance entry is in column 36 and must be Y, N, or blank.

---

**SYX5126      W CHARACTER SPECIFIED IN ENABLE DUP ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The enable dup entry is in column 34 and must be Y, N, or blank.

---

**SYX5128      W CHARACTER SPECIFIED IN SELF-CHECK ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The self-check entry is in column 30 and must be T, E, or blank.

---

**SYX5129      W CHARACTER SPECIFIED IN DATA TYPE ENTRY IS INVALID. ALPHAMERIC IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The data type entry is in column 27 and must be A, N, B, S, K, R, Z, D, E, F, O, or blank. A is assumed.

---

**SYX5130      W FIRST POSITION IN HIGH INTENSITY ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The high intensity entry is in columns 39 and 40.

---

**SYX5131      W FIRST POSITION IN HIGH INTENSITY ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The high intensity entry is in columns 39 and 40.

---

**SYX5132      W SECOND POSITION IN HIGH INTENSITY ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The high intensity entry is in columns 39 and 40. Column 39 contains a blank, and column 40 contains a zero or nonnumeric character.

---

**SYX5133**      **W INDICATOR SPECIFIED IN HIGH INTENSITY ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The high intensity entry is in columns 39 and 40. Valid indicator values are 01 through 99.

---

**SYX5134**      **W INDICATOR SPECIFIED IN HIGH INTENSITY ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The high intensity entry is in columns 39 and 40.

---

**SYX5135**      **W FIRST POSITION IN NONDISPLAY ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44.

---

**SYX5136**      **W FIRST POSITION IN NONDISPLAY ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44.

---

**SYX5137**      **W SECOND POSITION IN NONDISPLAY ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44. Column 43 contains a blank, and column 44 contains a zero or nonnumeric character.

---

**SYX5138**      **W INDICATOR SPECIFIED IN NONDISPLAY ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44. Valid indicator values are 01 through 99.

---

**SYX5139**      **W INDICATOR SPECIFIED IN NONDISPLAY ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44.

---

**SYX5140**      **W FIRST POSITION IN BLINK FIELD ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The blink field entry is in columns 41 and 42.

---

**SYX5141**      **W FIRST POSITION IN BLINK FIELD ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The blink field entry is in columns 41 and 42.

---

**SYX5142 W SECOND POSITION IN BLINK FIELD ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The blink field entry is in columns 41 and 42.

---

**SYX5143 W INDICATOR SPECIFIED IN BLINK FIELD ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The blink field entry is in columns 41 and 42. Valid indicator values are 01 through 99.

---

**SYX5144 W INDICATOR SPECIFIED IN BLINK FIELD ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The blink field entry is in columns 41 and 42.

---

**SYX5145 W FIRST POSITION IN REVERSE IMAGE ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The reverse image entry is in columns 45 and 46.

---

**SYX5146 W FIRST POSITION IN REVERSE IMAGE ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The reverse image entry is in columns 45 and 46.

---

**SYX5147 W SECOND POSITION IN REVERSE IMAGE ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The reverse image entry is in columns 45 and 46. Column 45 contains a blank, and column 46 contains a zero or nonnumeric character.

---

**SYX5148 W INDICATOR SPECIFIED IN REVERSE IMAGE ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The reverse image entry is in columns 45 and 46. Valid indicator values are 01 through 99.

---

**SYX5149 W INDICATOR SPECIFIED IN REVERSE IMAGE ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The reverse image entry is in columns 45 and 46.

---

**SYX5150 T NEITHER INPUT NOR OUTPUT WAS SPECIFIED FOR THIS FIELD.**

**Specification Type:** D

**Explanation:** Neither input (Y in column 26) nor output (Y or an indicator value in columns 23 and 24) was specified for the field being defined.

---

**SYX5151      W CHARACTER SPECIFIED IN CONSTANT TYPE ENTRY IS INVALID. BLANK IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The constant type entry is in column 56 and must be C, M, or blank.

---

**SYX5152      W CONSTANT TYPE ENTRY IS M. MIC DOES NOT START IN FIRST POSITION OF CONSTANT DATA ENTRY.**

**Specification Type:** D

**Explanation:** The constant type entry (column 56) is M, indicating that a message ID (MIC) should begin in the first position of the constant data entry (column 57). However, column 57 is blank. The constant data entry is ignored and a message ID must be specified by the user program when the format is displayed.

---

**SYX5153      W CONSTANT TYPE ENTRY IS M. MIC SPECIFIED IN CONSTANT DATA FIELD IS NOT NUMERIC AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The constant type entry (column 56) is M, indicating that the constant data entry contains a message ID (MIC) in columns 57 through 60. However, the value in columns 57 through 60 is nonnumeric. The constant data entry is ignored, and a message ID must be specified by the user program when the format is displayed.

---

**SYX5154      W CONSTANT TYPE ENTRY IS M. MESSAGE MEMBER ID FOLLOWING MIC IS INVALID AND DEFAULT IS ASSUMED.**

**Specification Type:** D

**Explanation:** The constant type entry (column 56) is M, indicating that the constant data entry contains a message ID (MIC) in columns 57 through 60 and a message member identifier in columns 61 and 62. However, an invalid message member identifier is in columns 61 and 62. The valid identifiers are U1, U2, P1, P2, M1, and M2. The invalid identifier is ignored and U1 is assumed.

---

**SYX5155      W FIELD LENGTH IS LESS THAN LENGTH OF CONSTANT DATA ENTRY. CONTINUATION ENTRY IS X BUT IS IGNORED.**

**Specification Type:** D

**Explanation:** Continuation was specified (X in column 80), but the field length entry (columns 15 through 18) is less than the length of the constant data entry (columns 57 through 79). The X in column 80 is ignored.

---

**SYX5157      W FIELD SPECIFIED AS OUTPUT-ONLY. DATA TYPE ENTRY IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** A data type entry (column 27) was specified for a field that was defined as output-only. A data type entry is invalid for output-only fields. However, an open (O) data type entry is allowed for output-only fields to allow processing of extended ideographic characters in the constant data entry.

---

**SYX5158      W FIELD SPECIFIED AS OUTPUT-ONLY. POSITION CURSOR ENTRY IS NOT BLANK OR N AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The position cursor entry is in columns 32 and 33 and must be blank or for an output-only field.

---

**SYX5159**      **W FIELD SPECIFIED AS OUTPUT-ONLY. MANDATORY ENTRY IS NOT BLANK OR N AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The mandatory entry is in column 29 and must be blank or N for an output-only field.

---

**SYX5160**      **W FIELD SPECIFIED AS OUTPUT-ONLY. MANDATORY FILL ENTRY IS NOT BLANK OR N AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The mandatory fill entry is in column 28 and must be blank or N for an output-only field.

---

**SYX5161**      **W FIELD SPECIFIED AS OUTPUT-ONLY. ADJUST/FILL ENTRY IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The adjust/fill entry is in column 31 and must be blank for an output-only field.

---

**SYX5162**      **T CHARACTER SPECIFIED IN SELF-CHECK ENTRY BUT FIELD LENGTH EXCEEDS 32.**

**Specification Type:** D

**Explanation:** A self-check entry (column 30) cannot be specified for any field with field length entry (columns 15 through 18) greater than 32.

---

**SYX5163**      **W FIELD SPECIFIED AS OUTPUT-ONLY. CONTROLLED FIELD EXIT ENTRY IS NOT BLANK OR N AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The controlled field exit entry is in column 35 and must be blank or N for an output-only field.

---

**SYX5164**      **W FIELD SPECIFIED AS OUTPUT-ONLY. AUTO RECORD ADVANCE ENTRY IS NOT BLANK OR N AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The auto record advance entry is in column 36 and must be blank or N for an output-only field.

---

**SYX5165**      **W FIELD SPECIFIED AS OUTPUT-ONLY. PROTECT FIELD ENTRY IS NOT BLANK OR N AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The protect field entry is in columns 37 and 38 and must be blank or N for an output-only field.

---

**SYX5166**      **W FIELD SPECIFIED AS OUTPUT-ONLY. ENABLE DUP ENTRY IS NOT BLANK OR N AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The enable dup entry is in column 34 and must be blank or N for an output-only field.

---

**SYX5167**      **W FIELD WAS SPECIFIED AS OUTPUT-ONLY. SELF-CHECK ENTRY IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The self-check entry is in column 30 and must be blank for an output-only field.

---

**SYX5168**      **W FIELD WAS SPECIFIED AS INPUT-ONLY. CONSTANT TYPE ENTRY IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The constant type entry is in column 56 and must be blank for an input-only field.

---

**SYX5169**      **W FIELD SPECIFIED AS INPUT-ONLY. CONSTANT DATA ENTRY IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The constant data entry is in columns 57 through 79 and must be blank for an input-only field.

---

**SYX5170**      **W FIELD SPECIFIED AS INPUT-ONLY. CONTINUATION ENTRY IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The continuation entry is in column 80 and must be blank for an input-only field.

---

**SYX5171**      **W FIRST POSITION IN UNDERLINE ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The underline entry is in columns 47 and 48.

---

**SYX5172**      **W FIRST POSITION IN UNDERLINE ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The underline entry is in columns 47 and 48.

---

**SYX5173**      **W SECOND POSITION IN UNDERLINE ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The underline entry is in columns 47 and 48. Column 47 contains a blank and column 48 contains a zero or nonnumeric character.

---

**SYX5174**      **W INDICATOR SPECIFIED IN UNDERLINE ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The underline entry is in columns 47 and 48. Valid indicator values are 01 through 99.

---

**SYX5175**      **W INDICATOR SPECIFIED IN UNDERLINE ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The underline entry is in columns 47 and 48.

---

**SYX5176**      **W CONSTANT TYPE ENTRY IS M. CONTINUATION ENTRY IS NOT BLANK AND IS IGNORED.**

**Specification Type:** D

**Explanation:** The continuation entry (column 80) must be blank if the constant type entry (column 56) is M.

---

**SYX5177**      **T LAST POSITION IN THIS FIELD IS BEYOND THE LAST POSITION ON THE SCREEN.**

**Specification Type:** D

**Explanation:** The field being defined extends beyond the end of the screen.

---

**SYX5178**      **T THIS RECORD FOLLOWS A CONTINUED RECORD BUT THE FORM TYPE ENTRY IS INVALID.**

**Specification Type:** D

**Explanation:** The form type entry (column 6) must be D for a record that follows a continued record (a record with X in column 80).

---

**SYX5179**      **T CONTINUATION ENTRY IS X BUT THIS IS THE LAST RECORD FOR THE FORMAT.**

**Specification Type:** D

**Explanation:** The continuation entry is in column 80.

---

**SYX5180**      **T CONSTANT DATA ENTRY IS AN SSP MESSAGE MEMBER BUT SSP-YES NOT GIVEN IN THE LOADMBR STATEMENT.**

**Specification Type:** D

**Explanation:** The message member identifier in columns 61 and 62 specifies an SSP message member (M1 or M2). However, M1 and M2 are valid only if SSP-YES is specified on the LOADMBR utility control statement.

---

**SYX5181**      **T CONSTANT TYPE ENTRY IS M BUT FIELD LENGTH ENTRY IS LESS THAN 6.**

**Specification Type:** D

**Explanation:** If M is specified in column 56 (constant type), then a 4-digit message identification code (MIC) is required in columns 57 through 60 and a 2-character message member identifier is required in columns 61 and 62. The field length entry is in columns 15 through 18, and the constant type entry is in column 56.

---

**SYX5182**      **T DATA TYPE ENTRY IS S OR D BUT FIELD LENGTH IS LESS THAN 2 OR GREATER THAN 16.**

**Specification Type:** D

**Explanation:** If data type is D (digits only), this message will not be issued when field length is less than 2. This message is issued when the field length entry of the D-spec is less than 2 (for S data type) or greater than 16 (for S or D data type). The field length entry is in columns 15 through 18, and the data type entry is in column 27.

---

---

**SYX5183      T MORE THAN 255 FORMATS HAVE BEEN SPECIFIED FOR THIS LOAD MEMBER.**

**Specification Type:** D

**Explanation:** A display format load member can contain up to 255 individual display formats.

---

**SYX5184      W MANDATORY FILL ENTRY IS Y AND ADJUST/FILL ENTRY IS Z OR B. ADJUST/FILL ENTRY IS IGNORED.**

**Specification Type:** D

**Explanation:** Mandatory fill (column 28) and adjust/fill (column 31) cannot both be specified for a field.

---

**SYX5185      T NUMBER OF LINES TO CLEAR PLUS START LINE NUMBER EXCEEDS THE NUMBER OF LINES ON THE SCREEN.**

**Specification Type:** S

**Explanation:** The number of lines to clear entry is in columns 19 and 20, and the start line number entry is in columns 17 and 18.

---

**SYX5186      T NUMBER OF INPUT FIELDS IN THIS FORMAT EXCEEDS MAXIMUM ALLOWED.**

**Specification Type:** D

**Explanation:** You can use the following equation to determine the maximum number of input fields allowed:

Maximum number of input fields = (255 minus the number of fields specified out of sequence minus the number of modulus 10 and modulus 11 fields {T or E in column 30} minus the number of secure operator ID fields minus one-half the length of the longest operator ID field) divided by 2.

---

**SYX5187      W CHARACTER SPECIFIED IN RETURN INPUT ENTRY IS INVALID. YES IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The return input entry is in column 22 and must be Y, N, or blank.

---

**SYX5188      W NONDISPLAY ENTRY AND BLINK FIELD ENTRY ARE BOTH Y. BLINK FIELD ENTRY IS IGNORED.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44, and the blink field entry is in columns 41 and 42.

---

**SYX5189      W NONDISPLAY ENTRY AND UNDERLINE ENTRY ARE BOTH Y. UNDERLINE ENTRY IS IGNORED.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44, and the underline entry is in columns 47 and 48.

---

**SYX5190**      **W NONDISPLAY ENTRY AND HIGH INTENSITY ENTRY ARE BOTH Y. HIGH INTENSITY ENTRY IS IGNORED.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44, and the high intensity entry is in columns 39 and 40.

---

**SYX5191**      **W NONDISPLAY ENTRY AND REVERSE IMAGE ENTRY ARE BOTH Y. REVERSE IMAGE ENTRY IS IGNORED.**

**Specification Type:** D

**Explanation:** The nondisplay entry is in columns 43 and 44, and the reverse image entry is in columns 45 and 46.

---

**SYX5192**      **T UNDERLINE, REVERSE IMAGE, AND HIGH INTENSITY ENTRIES ARE Y. ONLY TWO MAY BE GIVEN FOR ANY FIELD.**

**Specification Type:** D

**Explanation:** The underline entry is in columns 47 and 48, the reverse image entry is in columns 45 and 46, and the high intensity entry is in columns 39 and 40.

---

**SYX5193**      **T NO VALID SOURCE RECORDS WERE CONTAINED IN THIS SOURCE MEMBER.**

**Specification Type:** Not applicable

**Explanation:** Only comments were contained in this source member.

---

**SYX5194**      **W CONSTANT TYPE ENTRY IS C. OUTPUT DATA ENTRY IS AN INDICATOR BUT YES IS ASSUMED.**

**Specification Type:** D

**Explanation:** A constant type entry (column 56) of C is invalid if an indicator is specified in the output data entry (columns 23 and 24).

---

**SYX5195**      **W FIRST POSITION IN ERASE INPUT FIELDS ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The erase input fields entry is in columns 31 and 32.

---

**SYX5196**      **W FIRST POSITION IN ERASE INPUT FIELDS ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The erase input fields entry is in columns 31 and 32.

---

**SYX5197**      **W SECOND POSITION IN ERASE INPUT FIELDS ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The erase input fields entry is in columns 31 and 32. Column 31 contains a blank, and column 32 contains a zero or nonnumeric character.

---

**SYX5198**      **W INDICATOR SPECIFIED IN ERASE INPUT FIELDS ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The erase input fields entry is in columns 31 and 32. Valid indicator values are 01 through 99.

---

**SYX5199**      **W INDICATOR SPECIFIED IN ERASE INPUT FIELDS ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The erase input fields entry is in columns 31 and 32.

---

**SYX5200**      **W FIRST POSITION IN OVERRIDE FIELDS ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The override fields entry is in columns 33 and 34.

---

**SYX5201**      **W FIRST POSITION IN OVERRIDE FIELDS ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The override fields entry is in columns 33 and 34.

---

**SYX5202**      **W SECOND POSITION IN OVERRIDE FIELDS ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The override fields entry is in columns 33 and 34. Column 33 contains a blank, and column 34 contains a zero or nonnumeric character.

---

**SYX5203**      **W INDICATOR SPECIFIED IN OVERRIDE FIELDS ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The override fields entry is in columns 33 and 34. Valid indicator values are from 01 through 99.

---

**SYX5204**      **W INDICATOR SPECIFIED IN OVERRIDE FIELDS ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The override fields entry is in columns 33 and 34.

---

**SYX5205**      **W FIRST POSITION IN SUPPRESS INPUT ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The suppress input entry is in columns 35 and 36.

---

**SYX5206**      **W FIRST POSITION IN SUPPRESS INPUT ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The suppress input entry is in columns 35 and 36.

---

**SYX5207**      **W SECOND POSITION IN SUPPRESS INPUT ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The suppress input entry is in columns 35 and 36. Column 35 contains blank, and column 36 contains a zero or nonnumeric character.

---

**SYX5208**      **W INDICATOR SPECIFIED IN SUPPRESS INPUT ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The suppress input entry is in columns 35 and 36. Valid indicator values are 01 through 99.

---

**SYX5209**      **W INDICATOR SPECIFIED IN SUPPRESS INPUT ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The suppress input entry is in columns 35 and 36.

---

**SYX5210**      **W ENABLE FUNCTION KEYS ENTRY IS NOT Y, N, R OR BLANK. ENTRY IS IGNORED.**

**Specification Type:** S

**Explanation:** The enable function key's entry is in columns 27.

---

**SYX5211**      **W ENABLE COMMAND KEYS ENTRY IS NOT Y, N, R OR BLANK. ENTRY IS IGNORED.**

**Specification Type:** S

**Explanation:** The enable command key entry is in column 28.

---

**SYX5212**      **W KEY MASK ENTRY CONTAINS A FUNCTION KEY NUMBER, BUT THE ENABLE FUNCTION KEYS ENTRY IS NOT Y OR N.**

**Specification Type:** S

**Explanation:** The key mask entry is in columns 64 through 79. The enable function key's entry is in column 27.

---

**SYX5213**      **W KEY MASK ENTRY CONTAINS A COMMAND KEY CHARACTER, BUT THE ENABLE COMMAND KEYS ENTRY IS NOT Y OR N.**

**Specification Type:** S

**Explanation:** The key mask entry is in columns 64 through 79; the enable command key entry is in column 28.

---

**SYX5214**      **W KEY MASK ENTRY CONTAINS AN INVALID NUMBER OR CHARACTER. ENTRY IS IGNORED.**

**Specification Type:** S

**Explanation:** The key mask entry is in columns 64 through 79.

---

**SYX5215      W KEY MASK ENTRY CONTAINS AN EMBEDDED BLANK. NUMBER OR CHARACTER FOLLOWING BLANK IS IGNORED.**

**Specification Type:** S

**Explanation:** The key mask entry is in columns 64 through 79.

---

**SYX5216      T DATA TYPE ENTRY IS R, BUT FIELD LENGTH ENTRY IS GREATER THAN 128.**

**Specification Type:** D

**Explanation:** The data type entry is in column 27, and the field length entry is in columns 15 through 18. The R data type is used for data read from the magnetic stripe reader and cannot have a field length greater than 128.

---

**SYX5217      W DATA TYPE ENTRY IS R, BUT A FIELD ATTRIBUTE OTHER THAN NONDISPLAY WAS SPECIFIED. NONDISPLAY ASSUMED.**

**Specification Type:** D

**Explanation:** The data type entry is in column 27. The R data type is used for data read from the magnetic stripe reader and must be a nondisplay field.

---

**SYX5219      T THIS FIELD IS DEFINED AS IDEOGRAPHIC CODE CAPABLE BUT HAS LENGTH LESS THAN 4.**

**Specification Type:** D

**Explanation:** The input data type of the field specifies ideographic character input, but the field is not long enough to contain an ideographic character.

---

**SYX5220      T THIS FIELD IS IDEOGRAPHIC CODE CAPABLE BUT HAS AN ODD FIELD LENGTH.**

**Specification Type:** D

**Explanation:** The data type of the field specifies ideographic character input, which requires an even number of bytes, but the field length is odd.

---

**SYX5221      W FIELDS WITH SPECIFIED DATA TYPE SHOULD BE CONTAINED ON ONE LINE.**

**Specification Type:** D

**Explanation:** The field is an ideographic field extending to a second display line and beginning on an odd column, thus causing a character to be split between column 80 of one line and column 1 of the next line.

---

**SYX5222      T THIS FIELD COULD CAUSE AN IDEOGRAPHIC CHARACTER TO BE KEYED ACROSS 2 LINES.**

**Specification Type:** D

**Explanation:** The field is an ideographic field extending to a second display line and beginning on an odd column, thus causing a character to be split between column 80 of one line and column 1 of the next line.

---

**SYX5223      W THIS FIELD IS IDEOGRAPHIC CODE CAPABLE AND HAS SELF-CHECK. SELF-CHECK IS IGNORED.**

**Specification Type:** D

**Explanation:** The field is an ideographic data field and has modulus 10/11 self-check specified. The self-check specification is ignored.

---

**SYX5224      W DUP MAY NOT BE ALLOWED FOR THE SPECIFIED DATA TYPE.**

**Specification Type:** D

**Explanation:** This field is an open data type and dup key allowed is specified. Dup key allowed is ignored.

---

**SYX5225      W ADJUST/FILL MAY NOT BE ALLOWED FOR THE SPECIFIED DATA TYPE.**

**Specification Type:** D

**Explanation:** The ADJUST/FILL function will be performed only when the field is displayed at a nonideographic code capable work station.

---

**SYX5226      W THIS FIELD IS IDEOGRAPHIC CODE CAPABLE AND ZERO FILL IS SPECIFIED.**

**Specification Type:** D

**Explanation:** The field is capable of ideographic input and has zero fill specified. The field will be blank filled when in ideographic input mode.

---

**SYX5227      W THIS LINE CONTAINS AN IDEOGRAPHIC CHARACTER THAT MIGHT SPAN TWO DISPLAY LINES.**

**Specification Type:** D

**Explanation:** An ideographic character will not display properly when split between one display line and the next.

---

**SYX5228      T THE CONSTANT-MIC FIELD OF THIS SPECIFICATION ENDS IN IDEOGRAPHIC MODE.**

**Specification Type:** D

**Explanation:** A shift in character is missing at the end of the CONSTANT-MIC field on the D-specification.

---

**SYX5229      W THIS LINE CONTAINS AN EXTENDED IDEOGRAPHIC CHARACTER THAT WILL DISPLAY AS THE DEFAULT CHARACTER.**

**Specification Type:** D

**Explanation:** An extended ideographic character occurs in a field that does not begin with a shift out character; therefore, a special character will appear in place of this ideographic character when it is displayed.

**Note:** The constant data will be scanned for extended ideographic characters only if one of the following occurs:

1. The field is defined as an open field (data type O).
2. The field is not an open field and the first character of the constant data entry is a shift out character.

---

**SYX5230 T THIS FIELD CONTAINS AN IDEOGRAPHIC STRING WITH AN ODD LENGTH.**

**Specification Type:** D

**Explanation:** There is an ideographic string in the CONSTANT-MIC field of the D-specification, but there is an odd number of characters between the SO (shift out) and SI (shift in) delimiting the string.

---

**SYX5231 LOAD MEMBER NOT GENERATED.**

**Specification Type:** Not applicable

**Explanation:** The screen format load member could not be generated because terminal errors were encountered.

---

**SYX5232 W DATA TYPE SPECIFIED MAY BE INVALID.**

**Specification Type:** D

**Explanation:** The data type specified (O) may be invalid because of nonideographic SSP.

---

**SYX5235 T INVALID FIRST CHARACTER IN HELP DISPLAY SCREEN FORMAT NAME. MUST BE ALPHABETIC, @, #, OR \$.**

**Specification Type:** H

**Explanation:** The help display screen format name is in columns 7 through 14.

---

**SYX5236 T HELP DISPLAY SCREEN FORMAT NAME CONTAINS A BLANK. NAME MUST BE EXACTLY 8 CHARACTERS.**

**Specification Type:** H

**Explanation:** The help display screen format name is in columns 7 through 14.

---

**SYX5237 T HELP DISPLAY SCREEN FORMAT NAME CONTAINS A QUOTE OR A COMMA.**

**Specification Type:** H

**Explanation:** The help display screen format name is in columns 7 through 14.

---

**SYX5238 T THE LAST TWO POSITIONS OF THE HELP DISPLAY SCREEN FORMAT NAME ARE NOT NUMERIC.**

**Specification Type:** H

**Explanation:** The last two positions of the help display screen format name are used for roll key processing while the help format is displayed. The last two positions must contain numeric characters only. The help display screen format name is in columns 7 through 14.

---

**SYX5239 T HELP DISPLAY SCREEN LIBRARY NAME SPECIFIED BUT NO LOAD MEMBER NAME SPECIFIED.**

**Specification Type:** H

**Explanation:** If the help display screen library name is specified, the help display screen load member name must also be specified. The help display screen load member name is in columns 16 through 23. The help display screen library name is in columns 25 through 32.

---

**SYX5240 T INVALID FIRST CHARACTER IN HELP DISPLAY SCREEN LOAD MEMBER NAME. FIRST CHARACTER MUST BE ALPHABETIC, @, #, OR \$.**

**Specification Type:** H

**Explanation:** The help display screen load member name is in columns 16 through 23.

---

**SYX5241 T HELP DISPLAY SCREEN LOAD MEMBER NAME CONTAINS AN EMBEDDED BLANK.**

**Specification Type:** H

**Explanation:** The help display screen load member name is in columns 16 through 23.

---

**SYX5242 T HELP DISPLAY SCREEN LOAD MEMBER NAME CONTAINS A QUOTE, COMMA, OR PERIOD.**

**Specification Type:** H

**Explanation:** The help display screen load member name is in columns 16 through 23.

---

**SYX5243 T INVALID FIRST CHARACTER IN HELP DISPLAY SCREEN LIBRARY NAME. FIRST CHARACTER MUST BE ALPHABETIC, @, #, OR \$.**

**Specification Type:** H

**Explanation:** A library name can be up to 8 characters long and must begin with an alphabetic character (A through Z, @, # or \$). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), single quotes ('), blanks, question marks (?), slashes (/), and hyphens (-). Do not use F1, READER, PRINT or DISK as a library name. The help display screen library name is in columns 25 through 32.

---

**SYX5244 T HELP DISPLAY SCREEN LIBRARY NAME CONTAINS AN EMBEDDED BLANK.**

**Specification Type:** H

**Explanation:** The help display screen library name is in columns 25 through 32.

---

**SYX5245 T HELP DISPLAY SCREEN LIBRARY NAME CONTAINS AN INVALID CHARACTER.**

**Specification Type:** H

**Explanation:** The help display screen library name is in columns 25 through 32.

---

**SYX5246 T HELP DISPLAY SCREEN LOAD MEMBER NAME IS INVALID. THE NAME CANNOT BE ALL, DIR, SYSTEM OR NEW.**

**Specification Type:** H

**Explanation:** Load member names of ALL, DIR, SYSTEM or NEW are invalid names. The help display screen load member name is in columns 16 through 23.

---

**SYX5247 T HELP DISPLAY SCREEN LIBRARY NAME IS INVALID. THE NAME CANNOT BE F1, READER, PRINT OR DISK.**

**Specification Type:** H

**Explanation:** Library names of F1, READER, PRINT, or DISK are invalid names. The help display screen library name is in columns 25 through 32.

---

**SYX5248 T MAXIMUM NUMBER OF HELP AREA DEFINITIONS EXCEEDED.****Specification Type:** H

**Explanation:** This format contains too many H-specifications. The format object code generated for all the specified help area definitions exceeds the maximum allowed storage size. To reduce the amount of storage needed, you must do either or both of the following:

- Reduce the number of H-specifications
- Remove load member names or library names from the H-specifications wherever possible

---

**SYX5249 T H SPECIFICATION FOUND BEFORE FIRST S SPECIFICATION IN THE SOURCE MEMBER.****Specification Type:** H

**Explanation:** An H-specification was found prior to finding an S-specification in the source member. The first record coded in the display format source specifications must be the screen display control specification (the S-specification).

---

**SYX5250 T VALUE SPECIFIED FOR HELP AREA TOP ROW NUMBER EXCEEDS THE NUMBER OF LINES ON THE SCREEN.****Specification Type:** H

**Explanation:** The help area upper left row number entry is in columns 34 and 35.

---

**SYX5251 T SECOND POSITION OF HELP AREA TOP ROW NUMBER ENTRY IS ZERO OR IS NOT NUMERIC.****Specification Type:** H

**Explanation:** The help area upper left row number entry is in columns 34 and 35. Column 34 contains a blank, and column 35 contains a zero or a nonnumeric character.

---

**SYX5252 T VALUE SPECIFIED FOR HELP AREA TOP ROW NUMBER IS NOT NUMERIC.****Specification Type:** H

**Explanation:** The help area upper left row number entry is in columns 34 and 35.

---

**SYX5253 T VALUE SPECIFIED FOR HELP AREA TOP ROW NUMBER IS ZERO.****Specification Type:** H

**Explanation:** The help area upper left row number entry is in columns 34 and 35.

---

**SYX5254 T HELP AREA TOP ROW NUMBER PLUS START LINE NUMBER EXCEEDS NUMBER OF LINES ON THE SCREEN.****Specification Type:** H

**Explanation:** The help area upper left row number (specified in columns 34 and 35 on the H-specification) plus the start line number (specified in columns 17 and 18 on the S-specification) exceeds the number of lines on the screen. \$SFGR assumes the screen contains 24 lines for an 80-column format and 27 lines for a 132-column format.

---

**SYX5255**      **T VALUE SPECIFIED FOR HELP AREA LEFT COLUMN NUMBER EXCEEDS THE NUMBER OF HORIZONTAL POSITIONS ON THE SCREEN.**

**Specification Type:** H

**Explanation:** The help area upper left column number entry is in columns 36 and 37.

---

**SYX5256**      **T SECOND POSITION OF HELP AREA LEFT COLUMN NUMBER ENTRY IS ZERO OR IS NOT NUMERIC.**

**Specification Type:** H

**Explanation:** The help area upper left column number entry is in columns 36 and 37. Column 36 contains a blank, and column 37 contains a zero or a nonnumeric character.

---

**SYX5257**      **T VALUE SPECIFIED FOR HELP AREA LEFT COLUMN NUMBER IS NOT NUMERIC.**

**Specification Type:** H

**Explanation:** The help area upper left column number entry is in columns 36 and 37.

---

**SYX5258**      **T VALUE SPECIFIED FOR HELP AREA LEFT COLUMN NUMBER IS ZERO.**

**Specification Type:** H

**Explanation:** The help area upper left column number entry is in columns 36 and 37.

---

**SYX5259**      **T ONLY ONE PAIR OF ROW AND COLUMN NUMBERS IS SPECIFIED. BOTH PAIRS MUST BE EITHER SPECIFIED OR OMITTED.**

**Specification Type:** H

**Explanation:** Either the upper left row and column numbers (specified in columns 34 through 37) for the help area are blank and the lower right row and column numbers (specified in columns 39 through 42) are specified, or the upper left row and column numbers are specified and the lower right row and column numbers are blank. Both the upper left and lower right row and column numbers must be specified, or they both must be left blank.

---

**SYX5260**      **T VALUE SPECIFIED FOR HELP AREA BOTTOM ROW NUMBER EXCEEDS THE NUMBER OF LINES ON THE SCREEN.**

**Specification Type:** H

**Explanation:** The help area lower right row number entry is in columns 39 and 40.

---

**SYX5261**      **T SECOND POSITION OF HELP AREA BOTTOM ROW NUMBER ENTRY IS ZERO OR IS NOT NUMERIC.**

**Specification Type:** H

**Explanation:** The help area lower right row number entry is in columns 39 and 40. Column 39 contains a blank and column 40 contains a nonnumeric character.

---

**SYX5262**      **T VALUE SPECIFIED FOR HELP AREA BOTTOM ROW NUMBER IS NOT NUMERIC.**

**Specification Type:** H

**Explanation:** The help area lower right row number entry is in columns 39 and 40.

---

**SYX5263**      **T VALUE SPECIFIED FOR HELP AREA BOTTOM ROW NUMBER IS ZERO.**

**Specification Type:** H

**Explanation:** The help area lower right row number entry is in columns 39 and 40.

---

**SYX5264**      **T HELP AREA BOTTOM ROW NUMBER PLUS START LINE NUMBER EXCEEDS NUMBER OF LINES ON THE SCREEN.**

**Specification Type:** H

**Explanation:** The help area lower right row number (specified in columns 39 and 40 on the H-specification) plus the start line number (specified in columns 17 and 18 on the S-specification) exceeds the number of lines on the screen. \$SFGR assumes the screen contains 24 lines for an 80-column format and 27 lines for a 132-column format.

---

**SYX5265**      **T VALUE SPECIFIED FOR HELP AREA RIGHT COLUMN NUMBER EXCEEDS THE NUMBER OF HORIZONTAL POSITIONS ON THE SCREEN.**

**Specification Type:** H

**Explanation:** The help area lower right column number entry is in columns 41 and 42.

---

**SYX5266**      **T SECOND POSITION OF HELP AREA RIGHT COLUMN NUMBER ENTRY IS ZERO OR IS NOT NUMERIC.**

**Specification Type:** H

**Explanation:** The help area lower right column number entry is in columns 41 and 42. Column 41 contains a blank, and column 42 contains a zero or a nonnumeric character.

---

**SYX5267**      **T VALUE SPECIFIED FOR HELP AREA RIGHT COLUMN NUMBER IS NOT NUMERIC.**

**Specification Type:** H

**Explanation:** The help area lower right column number entry is in columns 41 and 42.

---

**SYX5268**      **T VALUE SPECIFIED FOR HELP AREA RIGHT COLUMN NUMBER IS ZERO.**

**Specification Type:** H

**Explanation:** The help area lower right column number entry is in columns 41 and 42.

---

**SYX5269**      **T HELP AREA TOP ROW NUMBER IS GREATER THAN HELP AREA BOTTOM ROW NUMBER.**

**Specification Type:** H

**Explanation:** The row number specified for the upper left corner of the help area (in columns 34 and 35) is greater than the row number specified for the lower right corner of the help area (in columns 39 and 40). The upper left row number must be less than or equal to the lower right row number.

---

**SYX5270      T HELP AREA LEFT COLUMN NUMBER IS GREATER THAN HELP AREA  
RIGHT COLUMN NUMBER.**

**Specification Type:** H

**Explanation:** The column number specified for the upper left corner of the help area (in columns 36 and 37) is greater than the column number specified for the lower right corner of the help area (in columns 41 and 42). The upper left column number must be less than or equal to the lower right column number.

---

**SYX5271      W FIRST POSITION IN SUPPRESS INDICATOR ENTRY IS Y. BLANKS  
ARE ASSUMED.**

**Specification Type:** H

**Explanation:** The selection indicator entry is in columns 44 and 45. Column 44 contains a Y. An indicator number or blanks are the only valid entries.

---

**SYX5272      W FIRST POSITION IN SUPPRESS INDICATOR ENTRY IS N. BLANKS  
ARE ASSUMED.**

**Specification Type:** H

**Explanation:** The selection indicator entry is in columns 44 and 45. Column 44 contains an N. An indicator number or blanks are the only valid entries.

---

**SYX5273      W SECOND POSITION IN SUPPRESS INDICATOR ENTRY IS INVALID.  
BLANK IS ASSUMED.**

**Specification Type:** H

**Explanation:** The selection indicator entry is in columns 44 and 45. Column 44 contains a blank and column 45 contains a zero or nonnumeric character.

---

**SYX5274      W INDICATOR SPECIFIED IN SUPPRESS INDICATOR ENTRY IS  
INVALID. BLANKS ARE ASSUMED.**

**Specification Type:** H

**Explanation:** The selection indicator entry is in columns 44 and 45. Valid indicator values are from 01 through 99.

---

**SYX5275      W INDICATOR SPECIFIED IN SUPPRESS INDICATOR ENTRY IS 00.  
BLANKS ARE ASSUMED.**

**Specification Type:** H

**Explanation:** The selection indicator entry is in columns 44 and 45. Valid indicator values are from 01 through 99.

---

**SYX5276      W H SPECIFICATION FOUND AFTER FIRST D SPECIFICATION IN THE  
SOURCE MEMBER.**

**Specification Type:** H

**Explanation:** An H-specification was found after finding a D-specification in the source member. All H-specifications must be coded before the first D-specification in the source member. The H-specification is assumed to be a comment. The help area defined by the H-specification is not created.

---

**SYX5280      W FIRST POSITION IN RESTORE APPLICATION SCREEN ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** H

**Explanation:** The restore application screen entry is in columns 47 and 48.

---

**SYX5281      W FIRST POSITION IN RESTORE APPLICATION SCREEN ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** H

**Explanation:** The restore application screen entry is in columns 47 and 48.

---

**SYX5282      W SECOND POSITION IN RESTORE APPLICATION SCREEN ENTRY IS ZERO OR NOT NUMERIC. Y IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** H

**Explanation:** The restore application screen entry is in columns 47 and 48. Column 47 contains a blank, and column 48 contains a zero or nonnumeric character.

---

**SYX5283      W INDICATOR SPECIFIED IN RESTORE APPLICATION SCREEN ENTRY IS INVALID. Y IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** H

**Explanation:** The restore application screen entry is in columns 47 and 48. Valid indicator values are 01 through 99.

---

**SYX5284      W INDICATOR SPECIFIED IN RESTORE APPLICATION SCREEN ENTRY IS 00. Y IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** H

**Explanation:** The restore application screen entry is in columns 47 and 48. Valid indicator values are 01 through 99.

---

**SYX5285      W FIRST POSITION IN NULL FILL ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The null fill entry is in columns 37 and 38.

---

**SYX5286      W FIRST POSITION IN NULL FILL ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** S

**Explanation:** The null fill entry is in columns 37 and 38.

---

**SYX5287      W SECOND POSITION IN NULL FILL ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The null fill entry is in columns 37 and 38. Column 37 contains a blank and column 38 contains a zero or a nonnumeric character.

---

**SYX5288**      **W INDICATOR SPECIFIED FOR NULL FILL ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The null fill entry is in columns 37 and 38. Valid indicator values are 01 through 99.

---

**SYX5289**      **W INDICATOR SPECIFIED FOR NULL FILL ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** S

**Explanation:** The null fill entry is in columns 37 and 38. Valid indicator values are 01 through 99.

---

**SYX5290**      **W CHARACTER SPECIFIED IN LOWERCASE ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The lowercase entry is in column 51 and must be Y, N, or blank. N is assumed.

---

**SYX5291**      **W LOWERCASE SPECIFIED FOR OUTPUT ONLY FIELD IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The lowercase entry is in column 51. The lowercase entry must be blank or N for output fields.

---

**SYX5292**      **W LOWERCASE SPECIFIED FOR R BADGE READER FIELD IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The lowercase entry is in column 51. The lowercase entry must be blank or N for input fields with a data type of R (badge reader).

---

**SYX5293**      **W LOWERCASE SPECIFIED FOR NUMERIC, SIGNED NUMERIC, OR DIGITS ONLY FIELD IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** D

**Explanation:** The lowercase entry is in column 51. The lowercase entry must be blank or N for input fields with a data type of N (numeric only), S (signed numeric), or D (digits only). N is the default.

---

**SYX5294**      **W FIRST POSITION IN BOUNDARY INDICATOR ENTRY IS Y. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** H

**Explanation:** The boundary indicator entry is in columns 50 and 51.

---

**SYX5295**      **W FIRST POSITION IN BOUNDARY INDICATOR ENTRY IS N. SECOND POSITION IS NOT BLANK AND IS IGNORED.**

**Specification Type:** H

**Explanation:** The boundary indicator entry is in columns 50 and 51.

---

---

**SYX5296 W SECOND POSITION IN BOUNDARY INDICATOR ENTRY IS ZERO OR NOT NUMERIC. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** H

**Explanation:** The boundary indicator entry is in columns 50 and 51. Column 50 contains a blank, and column 51 contains a zero or a nonnumeric character.

---

**SYX5297 W INDICATOR SPECIFIED IN BOUNDARY INDICATOR ENTRY IS INVALID. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** H

**Explanation:** The boundary indicator entry is in columns 50 and 51. Valid indicator values are 01 through 99.

---

**SYX5298 W INDICATOR SPECIFIED IN BOUNDARY INDICATOR ENTRY IS 00. NO IS ASSUMED FOR THIS ENTRY.**

**Specification Type:** H

**Explanation:** The boundary indicator entry is in columns 50 and 51. Valid indicator values are 01 through 99.

---

**SYX5299 W CHARACTER SPECIFIED FOR 132 CAPABLE IS INVALID. NO IS ASSUMED.**

**Specification Type:** S

**Explanation:** The 132 capable field is in column 39 of the S specification. Valid entries are Y (yes), N (no), or blank.

---

**SYX5300 W ONLINE DOCUMENTATION ENTRY IS NOT Y, N, OR BLANK. ENTRY IS IGNORED.**

**Specification Type:** H

**Explanation:** The online documentation entry is in column 53.

---

**SYX5301 T INVALID FIRST CHARACTER IN HELP TEXT LABEL. MUST BE ALPHABETIC, @, #, OR \$.**

**Specification Type:** H

**Explanation:** A help text label can be up to 8 characters long and must begin with an alphabetic character (A through Z, @, #, or \$). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), single quotes ('), or blanks. The help text label is in columns 7 through 14.

---

**SYX5302 T HELP TEXT LABEL CONTAINS AN EMBEDDED BLANK.**

**Specification Type:** H

**Explanation:** The help text label is in columns 7 through 14.

---

**SYX5303 T HELP TEXT LABEL CONTAINS A QUOTE OR COMMA.**

**Specification Type:** H

**Explanation:** The help text label is in columns 7 through 14.

---

**SYX5304 T INVALID FIRST CHARACTER IN HELP DOCUMENT NAME. MUST BE ALPHABETIC, NUMERIC, @, #, OR \$.**

**Specification Type:** H

**Explanation:** A help document name can be up to 8 characters long and must begin with an alphabetic or numeric character (A through Z, 0 through 9, @, #, or \$). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except question marks (?), single quotes ('), periods (.), hyphens (-), commas (,), or blanks. Do not use ALL as a help document name. The help document name is in columns 16 through 23.

---

**SYX5305 T HELP DOCUMENT NAME CONTAINS AN EMBEDDED BLANK.**

**Specification Type:** H

**Explanation:** The help document name is in columns 16 through 23.

---

**SYX5306 T HELP DOCUMENT NAME CONTAINS AN INVALID CHARACTER.**

**Specification Type:** H

**Explanation:** The help document name is in columns 16 through 23.

---

**SYX5307 T HELP DOCUMENT NAME CONTAINS A PERIOD.**

**Specification Type:** H

**Explanation:** The help document name is in columns 16 through 23.

---

**SYX5308 T HELP DOCUMENT NAME IS INVALID. THE NAME CANNOT BE ALL.**

**Specification Type:** H

**Explanation:** The help document name is in columns 16 through 23.

---

**SYX5309 W INVALID ENTRY IN RESTORE APPLICATION SCREEN ENTRY. MUST BE BLANK. ENTRY IS IGNORED.**

**Specification Type:** H

**Explanation:** The restore application screen entry is in columns 47 and 48.

---

**SYX5310 W INVALID ENTRY IN BOUNDARY INDICATOR ENTRY. MUST BE BLANK. ENTRY IS IGNORED.**

**Specification Type:** H

**Explanation:** The boundary indicator entry is in columns 50 and 51.

---

**SYX5311 T INVALID FIRST CHARACTER IN HELP DOCUMENT FOLDER NAME. MUST BE ALPHABETIC, @, #, OR \$.**

**Specification Type:** H

**Explanation:** A folder name can be up to 8 characters long and must begin with an alphabetic character (A through Z), @, # or \$. The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), single quotes ('), blanks, question marks (?), slashes (/), hyphens (-), equal signs (=), and greater than signs (>). Do not use ALL or #LIBRARY as a folder name. The help document folder name is in columns 25 through 32.

---

**SYX5312 T HELP DOCUMENT FOLDER NAME CONTAINS AN EMBEDDED BLANK.****Specification Type:** H**Explanation:** The help document folder name is in columns 25 through 32.

---

**SYX5313 T HELP DOCUMENT FOLDER NAME CONTAINS AN INVALID CHARACTER.****Specification Type:** H**Explanation:** The help document folder name is in columns 25 through 32.

---

**SYX5314 T HELP DOCUMENT FOLDER NAME IS INVALID. THE NAME CANNOT BE ALL OR #LIBRARY.****Specification Type:** H**Explanation:** The help document folder name is in columns 25 through 32.

---

**SYX5315 W EXTENDED HELP FORMAT NAME IN EXTERNAL DISPLAY FILE NOT SUPPORTED. IT WILL BE IGNORED.****Specification Type:** H**Explanation:** This help display screen format name is an extended format name because it contains characters which are not valid for DDS. Valid alternate format names are generated during SFGR to DDS conversion for extended format names encountered in all S-specifications. In order for this H-specification to be used the extended format name which it specifies must be replaced with the corresponding valid alternate format name.

However, since the screen format referred to by this H-specification is contained in a different format load member, such a replacement is not possible. This H-specification will be ignored. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5316 W FIRST CHARACTER OF FIELD NAME DOES NOT FOLLOW DDS STANDARDS.****Specification Type:** D**Explanation:** When this field is converted to DDS, the conversion utility replaces the first character with a character that DDS allows. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5317 W DUPLICATE FIELD NAME ENCOUNTERED. SFGR TO DDS CONVERSION UTILITY WILL ALTER NAME FOR DDS PROCESSING.****Specification Type:** D**Explanation:** When this field is converted to DDS, the conversion utility replaces this field name with an internally generated name. Duplicate field names are not allowed in DDS. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5318 W FIELD NAME CHARACTERS (OTHER THAN THE FIRST CHARACTER) DO NOT FOLLOW DDS STANDARDS.****Specification Type:** D**Explanation:** When this field is converted to DDS, the conversion utility replaces any invalid characters with characters allowed by DDS. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5319      W FIELD NAME CONTAINS AN EMBEDDED BLANK. THIS NAME DOES NOT FOLLOW DDS NAMING CONVENTIONS.**

**Specification Type:** D

**Explanation:** When this field is converted to DDS, the conversion utility replaces any embedded blank with a character that DDS allows. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5320      W LOAD MEMBER NAME CONTAINS CHARACTERS THAT DO NOT FOLLOW DDS FILE NAMING CONVENTIONS.**

**Specification Type:** H

**Explanation:** When this name is converted to DDS, the conversion utility replaces any invalid characters with characters allowed by DDS. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5321      W LOAD MEMBER LIBRARY NAME CONTAINS CHARACTERS THAT DO NOT FOLLOW DDS LIBRARY NAMING CONVENTIONS.**

**Specification Type:** H

**Explanation:** When this name is converted to DDS, the conversion utility replaces any invalid characters with characters allowed by DDS. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5322      W DOCUMENT NAME CONTAINS CHARACTERS THAT DO NOT FOLLOW DDS DOCUMENT NAMING CONVENTIONS.**

**Specification Type:** H

**Explanation:** When this name is converted to DDS, the conversion utility replaces any invalid characters with characters allowed by DDS. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5323      W DOCUMENT FOLDER NAME CONTAINS CHARACTERS THAT DO NOT FOLLOW DDS FOLDER NAMING CONVENTIONS.**

**Specification Type:** H

**Explanation:** When this name is converted to DDS, the conversion utility replaces any invalid characters with characters allowed by DDS. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5324      W BLANK FIELD NAME ENCOUNTERED ON AN INPUT OR OUTPUT FIELD. THIS CONFLICTS WITH DDS STANDARDS.**

**Specification Type:** D

**Explanation:** DDS requires field names on all fields with input or output variables. The conversion utility will generate a valid field name for this field. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5325      W TEXT LABEL CONTAINS CHARACTERS THAT DO NOT FOLLOW DDS DOCUMENT TEXT LABELING CONVENTIONS.**

**Specification Type:** H

**Explanation:** When this name is converted to DDS, the conversion utility replaces any invalid characters with characters allowed by DDS. This warning message can be ignored if syntax checking is being performed for creation of the format on a System/36.

---

**SYX5326 W PRINT KEY REQUESTS ARE INCONSISTENT. DDS USES FIRST S SPECIFICATION PRINT KEY DEFINITION.****Specification Type:** S

**Explanation:** The DDS print key specification determining where control is to be transferred when the print key is depressed is a file level parameter. On the System/36 this determination was at the record level. As a result of the difference, inconsistent print key requests can result in different results between this system and the System/36.

---

**SYX5327 W RIGHT TO LEFT DISPLAY REQUESTS ARE INCONSISTENT. DDS ONLY USES FIRST S SPECIFICATION.****Specification Type:** S

**Explanation:** The DDS left to right/right to left specification is a file level parameter. On the System/36 this between field movement could be controlled on a record basis. As a result of the difference, inconsistent cursor movement can result between this system and the System/36.

---

**SYX5328 W FORMAT CONTAINS USER-SUPPLIED \*DDS WHICH WOULD BE IGNORED BY SYSTEM/36.****Specification Type:** (Any)

**Explanation:** The function specified by the \*DDS statement may not exist on System/36, or on a previous release of the AS/400 system. System/36 treats this as a comment and the DDS is ignored, but this loss of function could affect the display or application if transported back to System/36. If transported to an AS/400 system where an earlier release is installed, the display might not compile successfully. Avoid using SDA to modify this source member, because SDA may inadvertently move the \*DDS comment to another location, causing a difference in function or causing the DDS compile to fail. The \*DDS statement is not recommended for general use, but is recommended for inserting the FRCDTA keyword into the generated DDS. This must be done immediately after an S-specification defining a format, which must be immediately written to the display even when the display file is created with DFRWRT(\*YES). See the Create System/36 Display File (CRTS36DSPF) and the Create Display File (CRTDSPF) commands in the *CL Reference* book and the *DDS Reference* book for more information. Because other types of DDS may not be compatible with System/36 environment applications, all other uses of \*DDS cannot be predicted.

---

**SYX5330 W LAST CHARACTER PROCESSED WAS A SHIFT OUT. DDS WILL REPLACE IT WITH A BLANK.****Specification Type:** D

**Explanation:** A DBCS string is being processed but the only character in the string is a shift out. It is replaced by a blank when the format is converted to DDS.

---

**SYX5331 T DOCUMENT FOLDER NAME CONTAINS SYNTAX ERROR. CHECK FOLDER EXTENSION NAME.****Specification Type:** H

**Explanation:** A document folder name cannot contain more than one period, cannot begin or end with a period, and cannot contain more than three characters following a period. The help document folder name is in columns 25 through 32.

---

**SYX5396      W VALUE SPECIFIED IN LINE NUMBER ENTRY IS INVALID FOR THIS MENU. ERRORS MAY OCCUR.**

**Specification Type:** D

**Explanation:** When the S and D specifications are created for a menu using the BLDMENU procedure, line 22 is reserved for operator input (and conditionally, line 23 for input overflow); line 24 is reserved for system messages. The input and output data characteristics for the fields on a menu should *not* be changed from what the build menu processing defines them as. Only the fields that describe the menu options and the field attributes that control color may be updated. If line 23 is specified as an input line, then line 24 is conditionally reserved for input overflow and this message is issued. The line number entry is in columns 19 and 20 of the D-specification.

---

**SYX5397      W SELF-CHECKING IS SPECIFIED FOR A RIGHT-TO-LEFT FIELD. CHECK DIGIT REQUIRED AT BEGINNING OF FIELD.**

**Specification Type:** D

**Explanation:** The self-check entry (column 30 on the D-specification) contains either a T for modulus 10 or an E for modulus 11 self-checking. The self-check digit must always be entered in the rightmost position of a field, regardless of field type. Therefore, for a right-to-left field, the check digit must be the first digit entered.

---

**SYX5398      W ADJUST/FILL IS SPECIFIED FOR A RIGHT-TO-LEFT FIELD. IGNORED WHENEVER CURSOR DIRECTION IS RIGHT-TO-LEFT AT FIELD EXIT.**

**Specification Type:** D

**Explanation:** The adjust/fill entry (column 31 on the D-specification) contains either a Z for right-adjust, zero fill, or a B for right-adjust, blank fill. The adjust and fill occurs when the cursor direction is left-to-right, but not when the cursor direction is right-to-left. The REVERSE key can be used in a right-to-left field to change the cursor direction to left-to-right.

---

**SYX5399      W CHARACTER SPECIFIED IN RIGHT-TO-LEFT ENTRY IS INVALID. N IS ASSUMED.**

**Specification Type:** S

**Explanation:** The right-to-left screen entry is in column 40 and must be Y, N, or blank. N is assumed.



---

## Appendix E. IBM-Supplied Programs

This appendix describes the following IBM-supplied programs you can call from your applications:

- QEXCVTDV converts AS/400 device names to System/36 environment device names.
- QEXRCDBK sets the *record blocking for shared files* attribute.
- QEXRMVDE removes a workstation device from the System/36 environment configuration.
- QEXSHRO sets the *shared opens of files* attribute.

---

### QEXCVTDV

Call the QEXCVTDV program from any high-level language program to convert AS/400 device names to the corresponding System/36 environment device names, and vice versa.

The program requires these parameters:

- 2-character System/36 device ID
- 10-character AS/400 device name
- 1-character device type

If parameter 1 is not blank, the System/36 device ID is converted to the AS/400 device name and placed in parameter 2. The initial value of parameter 2 is ignored.

If parameter 1 is blank and parameter 2 is not blank, the AS/400 device name is converted to the System/36 device ID and placed in parameter 1.

If an unknown or invalid device name or ID is specified, the resulting name is set to blanks, and the device type (parameter 3) is set to 0. If both parameters 1 and 2 are blank, parameter 1 is considered an invalid device ID and parameter 2 is considered an invalid device name.

The device ID specified in the SYMID parameter of the WORKSTN OCL statement is not recognized by this program.

Parameter 3 is set by the program to one of the following values:

- 0 Unknown.
- 1 The device is a work station.
- 2 The device is a printer.
- 3 The device is a tape device.
- 4 The device is a diskette drive.

The QEXCVTDV program must be called from another program that can receive the values returned by it. QEXCVTDV cannot be called directly from a procedure, because the procedure cannot receive the values returned by the program. The CL program shown in the following example returns the values in the local data area (LDA), and can be used to interface between QEXCVTDV and a procedure. The System/36 device ID is taken from or placed in positions 1 and 2 of the LDA, the

AS/400 device name is taken from or placed in positions 3 through 12 of the LDA, and the device type is placed in position 13 of the LDA.

```
PGM
DCL &DEVID *CHAR 2
DCL &DEVNAME *CHAR 10
DCL &DEVTYPE *CHAR 1
CHGVAR &DEVID %SST(*LDA 1 2)
CHGVAR &DEVNAME %SST(*LDA 3 10)
CALL QSSP/QEXCVTDV (&DEVID &DEVNAME &DEVTYPE)
CHGVAR %SST(*LDA 1 2) &DEVID
CHGVAR %SST(*LDA 3 10) &DEVNAME
CHGVAR %SST(*LDA 13 1) &DEVTYPE
ENDPGM
```

---

## QEXRCDBK

The QEXRCDBK program sets the *record blocking for shared files* attribute. Use it to indicate whether record blocking should be used for sequentially processed shared files in jobs run during a session. See the CHGS36 command in the *System/36 Environment Programming* book for more information on the *record blocking for shared files* attribute.

The parameter for the QEXRCDBK program follows:

Blocking Option CHAR (1)

Specify one of the following values for this parameter:

- '1' specifies the use of record blocking for shared files processed by jobs run in this session.
- '2' specifies that record blocking should not be used for shared files processed by jobs run in this session.

This attribute takes effect when the next RUN OCL statement is processed.

---

## QEXRMVDE

QEXRMVDE removes a workstation device from the System/36 environment configuration.

The QEXRMVDE program has the following parameters:

Device Name CHAR (10)  
Return Code CHAR (1)

The return code parameter is optional. If it is not specified, the program issues a message instead of a return code. If a device name is not specified, \*SIGNOFF is assumed.

Specify one of the following values for parameter 1:

- \* Specifies that the device for the job in which QEXRMVDE is called is removed. This is valid only for interactive jobs. If the System/36 environment is not active in the job, the device is removed immediately from the configuration. If the System/36 environment is active, the device is removed when the System/36 environment ends.

### **\*SIGNOFF**

Specifies that the device for the job in which QEXRMVDE is called is removed when the user signs off. This is valid only for interactive jobs. This is the default value if the parameter is not specified.

### **name**

Specifies the 10-character name of the device that is to be removed. The device is removed immediately from the configuration.

Specify a 1-character variable for parameter 2. It will be set to one of the following return code values by QEXRMVDE:

- 0** The device was successfully removed from the configuration.
- 1** If possible, the device will be removed from the configuration either when the System/36 environment ends or when the user signs off.  
  
No additional error checking is done when this return code is set. There are several situations in which the device will not be removed later. A message will be placed in the job log at the time the deferred request is processed to indicate either that the device was removed, or why it was not removed.
- 2** The device does not exist, or is not configured in the System/36 environment; no action was taken.  
  
This return code can be set if either \* or a device name is specified. If \*SIGNOFF is specified, this condition is not checked because even if the device is not in the S/36 configuration at the time the program is called, it may be added before the user signs off. A message will be placed in the job log if the device is not in the configuration when the user signs off.
- 3** Either \* or \*SIGNOFF was entered for the first parameter, but is not valid because the job is not an interactive job.
- 4** The device specified in the first parameter is not a workstation.
- 5** The device specified in the first parameter cannot be removed because it is in use by another job that is running in the System/36 environment.  
  
If \*SIGNOFF is specified, or \* is specified while running in the System/36 environment, this condition is not checked at the time the program is called. If the device is still in use at the time the user signs off or the System/36 environment ends, a message is placed in the job log.

If the second parameter is specified, no messages are issued by the program. Table E-1 on page E-4 lists the return codes and the messages that correspond to them. These messages are located in the message file QSSPMSG.

Table E-1. Messages issued by QEXRMVDE

Return Code	Message ID	Message Type
0	SSP0210	COMP
1	SSP0211	COMP
2	SSP0212	ESCAPE*
3	SSP0213	ESCAPE
4	SSP0214	ESCAPE
5	SSP0215	ESCAPE*

**Note:** Messages SSP0212 and SSP0215 are sent as diagnostic messages if they are issued when processing a deferred request (for instance, if \*SIGNOFF is specified for parameter 1, but the device is not configured in the System/36 environment at the time the user signs off, or if \* or \*SIGNOFF is specified and the device is in use in a secondary or group job when the user signs off or ends the System/36 environment).

QEXRMVDE can be used in the following ways:

- Call it in an interactive job to remove the job's device from the configuration. The removal can be immediate, deferred until the user ends the System/36 environment, or deferred until the user signs off. QEXRMVDE can be called in several ways:
  - In an initial program for a job to remove the device at signoff time.
  - In a user-written function that defers the request until signoff.
  - In a user-written function that starts the System/36 environment, and then defers the request to remove the device until either the System/36 environment ends or until the user signs off.
- It can be called to remove any workstation device from the System/36 environment configuration. In this case, a program can be set up to periodically remove all devices from the configuration. Because the program does not remove a device if it is in use, this can safely be done without impacting jobs that are currently running.

If you specify the name of a device, you must have \*CHANGE authority to object QS36ENV in library #LIBRARY, object type \*S36. If you specify \* or \*SIGNOFF for the first parameter, no special authority is required. This API is shipped with \*USE public authority.

---

## QEXSHRO

The QEXSHRO program sets the *shared opens of files* attribute. Use it to indicate whether the System/36 environment should use automatic shared opens of files during jobs run in a session. See the CHGS36 command in the *System/36 Environment Programming* book for more information on the *shared opens of files* attribute.

The parameter for the QEXSHRO program follows:

Shared Opens CHAR (1)

Specify one of the following values for this parameter:

- '1' specifies the use of automatic shared opens of files during jobs run in this session.
- '2' specifies that automatic shared opens of files should not be used during jobs run in this session.

This attribute takes effect when the next FILE OCL statement is processed.



---

## Bibliography

The following books contain information you may need. The books are listed with their full title and order number. Except where otherwise indicated, each is an AS/400 system information unit.

### General AS/400-Related Books

- *Backup and Recovery – Advanced*, SC41-4305, provides programmers with information about the different media available to save and protect system data, as well as a description of how to record changes made to database files and how that information can be used for system recovery and activity report information.
- *Data Management*, SC41-4710, provides application programmers and system programmers with information about using files in application programs.
- *Diskette General Information Manual*, GA21-9182, describes diskettes and how to handle them. This is not an AS/400 system information.
- *Distributed Data Management*, SC41-4307, provides application programmers and programmers with information about remote file processing. It describes how to create a distributed data management (DDM) file, how to define a remote file to the AS/400 system DDM, what file utilities are supported through DDM, and the requirements of the AS/400 system DDM as related to other systems.
- *Printer Device Programming*, SC41-3713, provides data processing managers with information on how to control and understand printing: printing elements and concepts, printer file support, print spooling support, printer connectivity, Advanced Function Printing support, and printing with personal computers.
- *Software Installation*, SC41-4120, provides system operators and administrators with information on new programs and releases and how to install them.
- *AS/400 Advanced Series Handbook*, GA19-5486, AS/400 users with introductory information on the features and capabilities of the AS/400 system.
- *Client Access/400 for DOS and OS/2 Technical Reference*, SC41-3563, provides application programmers with technical information needed to do advanced configuration or tailoring of Client Access for OS/400 for a special operating environment.
- *Client Access/400 for DOS with Extended Memory User Guide*, SC41-3501, provides PC Support users with concepts and examples of how to use the Client Access for OS/400 functions in DOS.
- *Client Access/400 for OS/2 User Guide*, SC41-3521, provides users with personal computers attached to an AS/400 system with concepts and examples of how to use the Client Access for OS/400 functions in Operating System/2\* (OS/2\*) program.
- *System/36 Environment Programming*, SC41-4730, provides application programmers and programmers with information identifying the differences in the applications process in the System/36 environment on the AS/400 system. It also contains information about techniques to use when programming the AS/400 system in the System/36 environment.
- *CL Programming*, SC41-4721, provides application programmers and programmers with information on AS/400 programming topics, including a general discussion on objects and libraries, control language (CL) programming, controlling flow and communicating between programs, working with objects in CL programs, and creating CL programs, predefined and impromptu messages and message handling, and defining and creating user-defined commands and menus.
- *CL Reference*, SC41-4722, provides application and system programmers with a description of the AS/400 control language (CL) and its commands. Each command is defined, including its syntax diagram, parameters, default values, and keywords.
- *Work Management*, SC41-4306, provides programmers with information about how to create an initial work management environment and how to change it.
- *Publications Reference*, SC41-4003, is intended for all AS/400 users. It identifies and describes the information in the AS/400 library, and provides guidance on how, when, and where to use the online and printed information.
- *Query/400 Use*, SC41-4210, provides business professionals and programmers with detailed information about how to use Query/400 to get data from any database file.
- *Security – Reference*, SC41-4302, provides anyone assigned system security responsibilities with information about general security concepts and planning for security on the system.
- *AS/400 Licensed Internal Code Diagnostic Aids – Volume 1*, LY44-4900, provides software service representatives with reference information about error logs, dumps, and traces. It also provides information about available tools, documentation, and procedures needed to solve AS/400 programming problems.

- *SNA Japanese 3270PC Emulation Users's Guide*, SC21-9744, is for Japanese users and provides the display station operator, system programmer, or host system programmer with information on how to use the SNA Japanese 3270PC emulation, which is provided with the Japanese 5250PC program.
- *System Operation*, SC41-4203, provides information on the features and capabilities of the AS/400 system. It familiarizes the user with characteristics of the system and the various licensed programs used on the AS/400 system.
- *System Startup and Problem Handling*, SC41-4206, provides system operators and system administrators with information about how to use the system unit operator panel, send and receive messages, respond to error messages, start and stop the system, use the display station function keys, control devices, and also process and manage jobs on the system.
- *Planning for and Setting Up OfficeVision/400*, SH21-0695, provides office administrators with information about planning for and setting up OfficeVision for OS/400.
- *Using OfficeVision/400*, SH21-0697, provides OfficeVision for OS/400 users with detailed information on how to use OfficeVision for OS/400, including information on handling mail and calendars.
- *Using OfficeVision/400 Word Processing*, SH21-0701, provides OfficeVision for OS/400 users with information on how to use the word processing functions of OfficeVision for OS/400.
- *9404 Models 1xx and Bxx-Fxx Installation and Upgrade*, SY44-3920, describes how to configure your system and how to install the system programs.

## Programming Language and Utility Books

- *ADTS/400: Character Generator Utility*, SC09-1769, provides application programmers and programmers with information about using the Application Development Tools character generator utility (CGU) to create and maintain a double-byte character set (DBCS) on the AS/400 system.
- *ADTS/400: Data File Utility*, SC09-1773, provides application programmers, programmers, and help desk personnel with information about using the Application Development Tools data file utility to create programs to enter data into files, update files, inquire into files, and run DFU programs.
- *ADTS/400: Screen Design Aid*, SC09-1768, provides application programmers with information

about using the Application Development Tools screen design aid (SDA) to design, create, and maintain display formats and menus on the AS/400 system in the System/38 environment.

- *ADTS/400: Screen Design Aid for the System/36 Environment*, SC09-1893, provides application programmers and system operators with information on how to use the screen design aid (SDA) for developing displays, menus, and online information in the System/36 environment of the AS/400 system.
- *ADTS/400: Source Entry Utility*, SC09-1774, provides application programmers and help desk personnel with information about using the Application Development Tools source entry utility (SEU) to create and edit source members.
- *BGU User's Guide and Reference*, SC09-1408, provides application programmers, programmers, system administrators, and business and technical professionals with information about using AS/400 Business Graphics Utility (BGU) to create various types of charts.
- *BASIC User's Guide and Reference*, SC09-1823, provides application programmers with information needed to write, test, and maintain BASIC programs on the AS/400 system.
- *System/36-Compatible COBOL Reference Summary*, SX09-1287, provides application programmers with a description of the System/36-compatible COBOL commands and statements.
- *System/36-Compatible COBOL User's Guide and Reference*, SC09-1815, provides application programmers with information about using COBOL in the System/36 environment of the AS/400 system.
- *System/36-Compatible RPG II User's Guide and Reference*, SC09-1818, provides application programmers with information on how to design, code, enter, compile, test, and run RPG II programs. In addition, differences between compiling in the System/36 environment and the AS/400 system are described and explained.
- *System/36-to-AS/400 Work Station Utility Conversion Guide*, provides application programmers with information about converting Work Station Utility (WSU) programs on System/36 to RPG III programs on the AS/400 system. The book contains suggested techniques and examples.
- *IDDU Use*, SC41-3704, provides administrative secretaries, business professionals, and programmers with detailed information on how to use the OS/400 interactive data definition utility (IDDU) to describe data dictionaries, files, and records to your system.

## Communications Books

- *Alerts Support*, SC41-4413, provides system operators, programmers, and system administrators with information on how to generate and send alerts
- *APPN Support*, SC41-3407, provides programmers with information for defining or using OS/400 advanced peer-to-peer networking (APPN).
- *APPC Programming*, SC41-3443, provides application programmers with information for developing application programs that use OS/400 advanced program-to-program communications (APPC).
- *Asynchronous Communications Programming*, SC41-3444, provides application programmers with information which includes a description of asynchronous communications, configuration requirements, commands used to start a communications session, and programming considerations for the AS/400 system.
- *BSC Equivalence Link Programming*, SC41-3445, provides application and system programmers with the information needed to write programs using OS/400 binary synchronous communications equivalence link (BSCSEL) to communicate with a remote system.
- *SNA Distribution Services*, SC41-3410, provides system operators and system administrators with information about administering data communications applications on the AS/400 system.
- *Finance Communications Programming*, SC41-3449, provides application programmers, system operators, and system administrators with information on the OS/400 finance support program. It describes how finance support communicates with a control unit and how to set up finance support.
- *ICF Programming*, SC41-3442, provides application programmers with the information needed to write application programs that use AS/400 communications and the ICF file.
- *Intrasystem Communications Programming*, SC41-3447, provides application programmers with information for defining or using intrasystem communications support to develop communications between two application programs on the same system.
- *RJE Guide*, SC09-1373, provides system operators, application programmers, and programmers with information about using the Communications Utilities remote job entry (RJE) to submit jobs to an IBM host processor.
- *Retail Communications Programming*, SC41-3448, provides application programmers and system administrators with information on using the OS/400 retail support program. It describes how retail

support communicates with a controller and how to set up retail support.

- *SNA Upline Facility Programming*, SC41-3446, provides application and system programmers with programming information for using the OS/400 Systems Network Architecture (SNA) upline facility with the AS/400 system, describing how to set up the SNA upline facility, how to write application programs for the SNA upline facility, and the return codes the SNA upline facility can send to a program.
- *3270 Device Emulation Support*, SC41-3408, provides display station operators and system programmers with information about using 3270 device emulation, configuring the system for 3270 device emulation and setting up 3270 printer emulation, and creating system device configuration on the host system.
- *Local Device Configuration*, SC41-4121, provides system operators and system administrators with information on how to do an initial configuration and how to change that configuration.

## Programming and Debugging Material

- *IBM CGU Keyboard Template (Large Keyboard)*, SC09-1027, shows the command keys used with the character generator utility.
- *IBM CGU Keyboard Template (Small Template)*, SC09-1028, shows the command keys used with the character generator utility.
- *IBM Command Key Template*, GX21-9799, is an adhesive template to be used with 5251 display stations.
- *IBM Keyboard Template*, GX21-7929, shows the command keys used with the AS/400 system program products.
- *IBM RPG Debugging Template*, GX21-9129, is a template to help you debug RPG problems.
- *IBM WSU/\$SFGR Debugging Template*, GX21-7926, is a template to help you debug display format and WSU problems.
- The following are blank, plastic templates on which you can place information about the command keys used by your programs:
  - *IBM 5251 Models 1 and 11 and IBM 5252 Dual Display Station Keyboard Template*, GX21-9266
  - *IBM 5251 Display Station Models 2 and 12 Keyboard Template*, GX21-9327
  - *IBM 5291 Display Station Keyboard Template*, GX21-9410

- *IBM 5292 Color Display Station Keyboard Template*, GX21-9414
- *IBM 5292 Color Display Station Select Options*, GX21-9451, shows how to control the special features of the 5292 display.
- *System/36 to AS/400 Migration Aid User's Guide and Reference*, SC09-1166, provides system operators, application programmers, programmers, and data processing managers with information about using the System/36-to-AS/400 Migration Aid to move System/36 items to the AS/400 system using menus and displays, or commands.

## Migration Books

- *System/36 Migration Planning*, SC41-4152, provides application programmers, system administrators, and data processing managers with information to help migration of products and applications with the System/36 Migration Aid. It includes information for planning the details of migration and performing the functions of the Migration Aid.

---

# Index

## Special Characters

?C'value'? (length) expression 3-13  
?CD? (return code) expression 3-13  
?CLIB? (current library) expression 3-16  
?Cn? (parameter length) expression 3-12  
?DATE? (program date) expression 3-16  
?DATEFMT? (date format) expression 3-17  
?DEV'unit'? (device name) expression 3-17  
?F'A,name'? or ?F'A,name,date'? (actual file size) expression 3-18  
?F'S,name'? or ?F'S,name,date'? (file size) expression 3-17  
?F'T,name'? (file type) expression 3-19  
?FLIB? (files library) expression 3-19  
?L'position,length'? (local data area) expression 3-19  
?MENU? (current menu) expression 3-21  
?Mmic? or ?M'mic,position,length'? (message member) expression 3-20  
?MSGID? (message ID) expression 3-21  
?n? (parameter) expression 3-8  
?n? substitution expression 2-3  
?n'value'? (default parameter) expression 3-9  
?n'value'? substitution expression 2-3  
?nF'value'? (forced value parameter) expression 3-10  
?nF'value'? substitution expression 2-4  
?nR? (missing parameter) expression 3-11  
?nR? substitution expression 2-4  
?nR'mic'? (missing parameter message) expression 3-12  
?nR'mic'? substitution expression 2-4  
?nT'value'? (temporary value parameter) expression 3-9  
?nT'value'? substitution expression 2-4  
?PRINTER? (session printer) expression 3-21  
?PROC? (first level procedure) expression 3-22  
?R? (required parameter) expression 3-10  
?R'mic'? (required parameter message) expression 3-11  
?SFLIB? (session files library) expression 3-22  
?SLIB? (session library) expression 3-22  
?SYSLIST? (system list device) expression 3-22  
?TIME? (system time) expression 3-23  
?USER? (operator's user ID) expression 3-23  
?VOLID? or ?VOLID'location'? (diskette or tape volume ID) expression 3-23  
?WS? (display station ID) expression 3-24  
// \* (informational message) statement 3-44  
// \*\* (system console message) statement 3-45

/\* statement 5-107  
\$ARSP utility program A-3  
\$BICR utility program A-3  
\$BMENU utility program A-5  
\$BUILD utility program A-6  
\$COPY utility program A-7  
\$CPPE utility program A-28  
\$CZUT utility program A-28  
\$DCOPY utility program A-29  
\$DDST utility program A-29  
\$DELET utility program A-29  
\$DPGP utility program A-31  
\$DPGR utility program A-32  
\$DUPRD utility program A-32  
\$FBLD utility program A-33  
\$FREE utility program A-34  
\$HELP utility program A-35  
\$HIST utility program A-35  
\$IDSET utility program A-35  
\$IEDS utility program A-35  
\$IENBL utility program A-35  
\$INIT utility program A-35  
\$LABEL utility program A-36  
\$MAINT utility program A-37  
\$MGBLD utility program A-55  
\$MMST utility program A-56  
\$PACK utility program A-56  
\$PNLM utility program A-56  
\$POST utility program A-57  
\$PRCED utility program A-57  
\$PRCLT utility program A-57  
\$PRLST utility program A-57  
\$PRPWD utility program A-57  
\$PRUED utility program A-57  
\$PRUID utility program A-57  
\$PRURS utility program A-57  
\$PRUSV utility program A-58  
\$RENAM utility program A-58  
\$RREDT utility program A-58  
\$RRESC utility program A-58  
\$RRLST utility program A-58  
\$RRSAV utility program A-59  
\$RRSTR utility program A-59  
\$RRTED utility program A-59  
\$RRTLT utility program A-59  
\$SETCF utility program A-59  
\$SETCP utility program A-60  
\$SFGR utility program A-60  
\$SINCT utility program A-62  
\$SINDL utility program A-62  
\$SINR utility program A-62

**\$SVCASR utility program** A-62  
**\$TCOPY utility program** A-62  
**\$TINIT utility program** A-64  
**\$TMSERV utility program** A-64  
**\$UASC utility program** A-65  
**\$UASF utility program** A-65  
**\$XNLM utility program** A-66  
**\$XNSH utility program** A-66  
**\$XREST utility program** A-66  
**\$XSAVE utility program** A-66  
**\* (comment) statement** 3-7  
**#GCFR utility program** A-66  
**#GSORT utility program** A-66  
**#KASRT utility program** A-69  
**#STRUP1 procedure** 4-3  
**+ (continuation character)** 2-7

## Numerics

3270 emulation 4-86, 4-92, 4-95

## A

**ABEND OCL statement** 5-5  
**ACTIVE (running procedures) condition** 3-26  
**actual file size (?F'A,name'? or ?F'A,name,date?')**  
   expression 3-18  
**alert indicators, changing** A-3  
**ALERT procedure** 4-4  
**ALLOCATE OCL statement** 5-6  
**ALOCFLDR procedure** 4-4  
**ALOCLIBR procedure** 4-5, A-42  
**ALTERBSC procedure** 4-5  
**ALTERCOM procedure** 4-5, A-59  
**alternative indexes, creating** A-34  
**ALTERSDL procedure** 4-10  
**APAR procedure** 4-10  
**APPNINFO procedure** 4-10  
**ARCHIVE procedure** 4-11  
**\$ARSP utility program** A-3  
**ASCII code tables** C-1  
**ASM procedure** 4-11  
**ASMLOAD procedure** 4-11  
**ASMSAVE procedure** 4-11  
**ASSIGN control command** 6-3  
**assigning values** 3-47  
**asynchronous communications** 5-91  
**ATTR OCL statement** 5-8  
**attributes**  
   communications A-59  
   for USER parameter 4-153  
   procedure 2-9  
**AUTO procedure** 4-11  
**AUTO C procedure** 4-11  
**automatic response**  
   control statement 4-197

**automatic response (continued)**  
   programming considerations 4-201  
   severity level 5-65  
   source member 4-202  
   source statements 4-197  
   specification statement 4-199  
   values A-3

**available disk space (BLOCKS) condition** 3-27

## B

**BACKUP procedure** 4-14  
**BALPRINT procedure** 4-14  
**basic data exchange files** 4-293  
**BASIC procedure** 4-14  
**BASICP procedure** 4-15  
**BASICR procedure** 4-15  
**BASICS procedure** 4-15  
**BASLOAD procedure** 4-15  
**BASSAVE procedure** 4-15  
**BGUCHART procedure** 4-16  
**BGUDATA procedure** 4-16  
**BGUGRAPH procedure** 4-16  
**BGULOAD procedure** 4-17  
**BGUSAVE procedure** 4-17  
**\$BICR utility program** A-3  
**BLDFILE procedure** 4-17, A-33  
**BLDGRAPH procedure** 4-19  
**BLDINDEX procedure** 4-20, A-34  
**BLDLIBR procedure** 4-22, A-37  
**BLDMENU procedure** 4-25  
**BLOCKS (available disk space) condition** 3-27  
**\$BMENU utility program** A-5  
**braces** 1-3  
**brackets** 1-4  
**BSCCEL and RPG II Telecommunications** 5-87  
**BUILD procedure** 4-29  
**\$BUILD utility program** A-6

## C

**CACHE procedure** 4-29  
**calling procedures** 2-8  
**CANCEL control command** 6-3  
**CANCEL OCL statement** 5-13  
**CANCEL statement** 3-46  
**capitalized expressions** 1-3  
**CATALOG procedure** 4-29  
**CEND statements** A-47  
**CGU procedure** 4-39  
**CGULOAD procedure** 4-39  
**CGUSAVE procedure** 4-39  
**change communications attributes** A-59  
**CHANGE control command** 6-5  
**change library member information** A-43

change library or directory size A-42  
**CHANGE OCL statement** 5-14  
 characters in names, CL commands 2-22  
**CHGXLATE procedure** 4-40  
**CHNGEMEM procedure** 4-40, A-43  
**CL commands**  
 See Control Language commands  
**CLRPF procedure** 4-43  
**CNFIGICF procedure** 4-43  
**CNFIGSSP procedure** 4-43  
**CNFIGX25 procedure** 4-44  
**COBLOAD procedure** 4-44  
**COBOL procedure** 4-44  
**COBOLC procedure** 4-44  
**COBOLCG procedure** 4-47  
**COBOLG procedure** 4-47  
**COBOLONL procedure** 4-47  
**COBOLP procedure** 4-47  
**COBSAVE procedure** 4-48  
**COBSDA procedure** 4-48  
**COBSEU procedure** 4-48  
 coding rules, OCL 5-3  
 coding techniques and performance tips 2-24  
**COMM OCL statement** 5-16  
**commands**  
 See also control commands  
 See also Control Language (CL) commands  
 definition 1-1  
 procedure 2-2  
**commas** 1-5  
**comment (\*) statement** 3-7, 4-69, 4-200  
**comments, OCL** 5-4  
**communications**  
 attributes A-59  
 ENABLED condition 3-32  
**comparing, equal to (string1=string2)**  
 condition 3-40  
**comparing, greater than (string1>string2)**  
 condition 3-41  
**compatible statements** A-24  
**COMPILE OCL statement** 5-17  
**COMPRESS procedure** 4-49  
**concatenate values using substitution**  
 expressions 2-26  
**CONDENSE procedure** 4-49, A-43  
**conditional expressions**  
 ACTIVE (running procedures) 3-26  
 BLOCKS (available disk space) 3-27  
 comparing, equal to (string1=string2) 3-40  
 CONSOLE (system console) 3-27  
 DATAF1 (files and libraries on disk) 3-28  
 DATA11 (files on diskette) 3-29  
 DATAT (files on tape) 3-30  
 DSPLY (display station type) 3-31  
 ENABLED (enabled communications) 3-32  
 EVOKED (evoked procedures) 3-32

**conditional expressions** (*continued*)  
 FILELIBL (File Library List) 3-32  
 IF 3-25  
 INQUIRY (inquiry mode) 3-33  
 JOBQ (job queue) 3-33  
 LISTDONE (phone list completion) 3-34  
 LOAD (library load members) 3-34  
 MRTMAX (Multiple Requesting Terminals) 3-35  
 PROC (library procedure members) 3-36  
 SECURITY (password security) 3-36  
 SOURCE (library source members) 3-37  
 string1>string2 (comparing, greater than) 3-41  
 SUBR (library subroutine members) 3-38  
 SWITCH (switches) 3-39  
 SWITCHn (individual switches) 3-39  
 VOLID (diskette and tape volume IDs) 3-41  
**conditioning CL commands** 2-22  
**CONSOLE (system console) condition** 3-27  
**CONSOLE control command** 6-11  
**continuation character (+)** 2-7  
**continuing the lines of a procedure** 2-7  
**control commands**  
 ASSIGN 6-3  
 CANCEL 6-3  
 CHANGE 6-5  
 CONSOLE 6-11  
 description 6-1  
 HOLD 6-11  
 INFOMSG 6-12  
 JOBQ 6-12  
 MENU 6-14  
 MODE 6-15  
 MSG 6-15  
 OFF 6-18  
 POWER 6-19  
 PRTY 6-19  
 RELEASE 6-19  
 REPLY 6-20  
 RESTART 6-20  
 START 6-22  
 STATUS 6-24  
 STATUSF 6-30  
 STOP 6-32  
 TIME 6-34  
 VARY 6-34  
**control expressions**  
 See procedure control expressions  
**Control Language (CL) commands**  
 conditioning 2-22  
 handling errors 2-22  
 procedures 2-20  
 prompting for 2-22  
 special characters in names 2-22  
 substitution expressions 2-20  
**converting**  
 decimal to hexadecimal B-1

**converting** (*continued*)  
     records to blocks for disk files B-3  
**COPY and CEND statements** A-47  
**copy data to or from tape** A-63  
**copy members from library** A-45  
**copy members from one library to another** A-44  
**copy members to library** A-50  
**COPY statements** A-47  
**COPY utility control statement** 1-2  
**\$COPY utility program** A-7  
**COPYDATA procedure** 4-50, A-7  
**COPYDIAG procedure** 4-55  
**COPYI1 procedure** 4-55  
**copying disk files** A-7  
**copying files** A-4  
**COPYPRT files, file format** 4-59  
**COPYPRT procedure** 4-57  
**\$CPPE utility program** A-28  
**create a library** A-37  
**CREATE procedure** 4-66  
**create source or procedure members** A-39  
**creating files** A-33  
**CSALL procedure** 4-71  
**current library (?CLIB?) expression** 3-16  
**current menu (?MENU?) expression** 3-21  
**\$CZUT utility program** A-28

## D

**data file information** 3-4  
**data in files, defining** 1-13  
**data record parameters** 5-90  
**DATAF1 (files and libraries on disk) condition** 3-28  
**DATAI1 (files on diskette) condition** 3-29  
**DATAT (files on tape) condition** 3-30  
**date format (?DATEFMT?) expression** 3-17  
**DATE OCL statement** 5-19  
**DATE procedure** 4-71  
**\$DCOPY utility program** A-29  
**\$DDST utility program** A-29  
**DEALLOC OCL statement** 5-21  
**DEBUG OCL statement** 5-22  
**debugging procedures** 2-27  
**decimal and hexadecimal conversions** B-1  
**default parameter (?n'value?) expression** 3-9  
**DEFINEID procedure** 4-73  
**DEFINEPN procedure** 4-73  
**defining data in files** 1-13  
**DEFINLOC procedure** 4-74  
**DEFINX21 procedure** 4-74  
**DEFINX25 procedure** 4-75  
**DEFSUBD procedure** 4-75  
**\$DELET utility program** A-29  
**DELETE procedure** 4-76  
**DELNRD procedure** 4-80

**device name (?DEV'unit?') expression** 2-21, 3-17  
**DFA procedure** 4-80  
**DFU procedure** 4-80  
**DFULOAD procedure** 4-81  
**DFUSAVE procedure** 4-81  
**DICTLOAD procedure** 4-81  
**DICTSAVE procedure** 4-81  
**directory to utilities and statements** 4-1  
**DISABLE procedure** 4-81  
**disk files** 1-6  
**disk VTOC listings**  
     beginning columns 4-38  
     by library 4-33  
     by single name 4-34  
     description of items 4-34  
     field lengths 4-38  
     sample printouts 4-33  
**diskette**  
     and tape volume IDs (VOLID) condition 3-41  
     information 3-5  
     processing 1-9  
     VTOC listing 4-39  
**display formats** 1-11, 3-3  
**DISPLAY procedure** 4-82  
**display station**  
     changing and controlling 1-15  
     environment A-59  
     ID 4-174  
     ID (?WS?) expression 3-24  
     pass-through 4-174  
     type (DSPLY) condition 3-31  
**displaying messages or display formats** 3-3  
**DLSLOAD procedure** 4-82  
**DLSSAVE procedure** 4-82  
**DOCCNV procedure** 4-82  
**DOCPLOAD procedure** 4-82  
**DOCPSAVE procedure** 4-82  
**document description record fields** 4-269  
**double-byte character set** 4-70  
**\$DPGP utility program** A-31  
**\$DPGR utility program** A-32  
**DSPLOCKS procedure** 4-83  
**DSPLY (display station type) condition** 3-31  
**DSPMNU procedure** 4-83  
**DSPSYS procedure** 4-83  
**DSU procedure** 4-83  
**DSULOAD procedure** 4-85  
**DSUSAVE procedure** 4-85  
**DUMP procedure** 4-86

## E

**EBCDIC code tables** C-1  
**EDITNRD procedure** 4-86  
**ELSE expressions** 3-43

**EM3270 procedure** 4-86  
**ENABLE procedure** 4-89  
**enabled communications (ENABLED)**  
    **condition** 3-32  
**end document record fields** 4-280  
**end of data statement (\*)** 5-107  
**END utility control statement** 1-2  
**ENTER procedure** 4-89  
**ENTER# procedure** 4-91  
**entered parameters, testing** 2-6  
**EP3270 procedure** 4-92  
**EPDOWNL procedure** 4-92  
**EPLMRG procedure** 4-92  
**ERAP procedure** 4-93  
**ERR procedure** 4-93  
**errors on CL commands** 2-22  
**ES3270 procedure** 4-95  
**EVALUATE statement** 3-46  
**evaluating substitution expressions** 3-51  
**EVOKE OCL statement** 5-23  
**EVOKED (evoked procedures) Condition** 3-32  
**exchange tape files** A-63  
**expressions**  
    ?VOLID? or ?VOLID'location'? (diskette or tape  
    volume ID) 3-23  
    actual file size (?F'A,name'? or  
    ?F'A,name,date'?) 3-18  
    braces 1-3  
    brackets 1-4  
    capitalized 1-3  
    commas 1-5  
    current library (?CLIB?) 3-16  
    current menu (?MENU?) 3-21  
    date format (?DATEFMT?) 3-17  
    default parameter (?n'value'?) 3-9  
    device name (?DEV'unit'?) 2-21, 3-17  
    display station ID (?WS?) 3-24  
    ELSE 3-43  
    file size (?F'S,name'? or ?F'S,name,date'?) 3-17  
    file type (?F't,name'?) 3-19  
    files library (?FLIB?) 2-21, 3-19  
    first level procedure (?PROC?) 3-22  
    forced value parameter (?nF'value'?) 3-10  
    length (?C'value'?) 3-13  
    local data area (?L'position,length'?) 3-19  
    message ID (?MSGID?) 2-21, 3-21  
    message member (?Mmic? or  
    ?M'mic,position,length'?) 3-20  
    missing parameter (?nR?) 3-11  
    missing parameter message (?NR'mic'?) 3-12  
    operator's user ID (?USER?) 3-23  
    parameter (?n?) 3-8  
    parameter length (?Cn?) 3-12  
    parentheses 1-4  
    preventing displaying of informational  
    messages 2-26

**expressions (continued)**  
    procedure control 2-2, 3-1  
    program date (?DATE?) 3-16  
    required parameter (?R?) 3-10  
    required parameter (?R'mic'?) 3-11  
    return code (?CD?) 3-13  
    session files library (?SFLIB?) 2-21, 3-22  
    session library (?SLIB?) 3-22  
    session printer (?PRINTER?) 3-21  
    substitution 3-7  
    system list device (SYSLIST?) 3-22  
    system time (?TIME?) 3-23  
    temporary value (?n'value'?) 3-9  
    underscores 1-4  
**extended character files** 1-7  
**EXTRACT procedure** 4-97

## F

**\$FBLD utility program** A-33  
**file format for COPYPRT files** 4-59  
**file format for PRTFILE files** 4-268  
**file library list (FILELIBL) condition** 3-32  
**FILE OCL statement (for disk files)** 5-25  
**FILE OCL statement (for diskette files)** 5-33  
**FILE OCL statement (for tape files)** 5-37  
**file size (?F'S,name'? or ?F'S,name,date'?)**  
    **expression** 3-17  
**FILE statement** A-68  
**file type (?F'T,name'?) expression** 3-19  
**FILELIB OCL statement** 5-43  
**FILELIBL (File Library List) condition** 3-32  
**files and libraries on disk (DATAF1) condition** 3-28  
**files library (?FLIB?) expression** 2-21, 3-19  
**files on diskette (DATAI1) condition** 3-29  
**files on tape (DATAT) condition** 3-30  
**finance communications** 5-91  
**first level procedure (?PROC?) expression** 3-22  
**fixed-format menu, definition** 4-25  
**FLIB procedure** 4-98  
**folders and folder members** 1-9  
**forced value parameter expression (?nF'value'?)**  
    **expression** 3-10  
**FORMAT procedure** 4-99  
**FORMS OCL statement** 5-46  
**FORTRAN procedures** 4-103  
**\$FREE utility program** A-34  
**free-format menu, definition,** 4-25  
**FROMLIBR procedure** 4-103, A-45

## G

**#GCFR utility program** A-66  
**general format record fields** 4-272  
**GOTO and TAG statements** 3-52

#GSORT utility program A-66

## H

header record fields 4-268  
HELP procedure 4-109  
\$HELP utility program A-35  
hexadecimal and decimal conversions B-1  
HISTCOPY procedure 4-114  
HISTCRT procedure 4-114  
history file 2-9  
HISTORY procedure 4-115  
HOLD control command 6-11

## I

I-exchange files 4-293  
IBM-supplied programs E-1  
ICFDEBUG procedure 4-115  
ICVERIFY procedure 4-115  
IDDU procedure 4-115  
IDDUDCT procedure 4-116  
IDDUDFN procedure 4-116  
IDDUDISK procedure 4-116  
IDDULINK procedure 4-116  
IDDUPRT procedure 4-117  
IDDURBLD procedure 4-118  
IDDUXLAT procedure 4-119  
\$IDSET utility program A-35  
\$IEDS utility program A-35  
\$IENBL utility program A-35  
IF conditional expression  
    definition 3-25  
    statement portion 3-43  
    using 2-6  
IGC procedure 4-119  
IMAGE OCL statement 5-49  
INCLUDE OCL statement 5-49  
indexes, alternative A-34  
individual switches (SWITCHn) condition 3-39  
INFOMSG control command 6-12  
INFOMSG OCL statement 5-51  
informational message (// \*) statement 3-44  
INIT procedure 4-119  
\$INIT utility program A-35  
INIT9332 procedure 4-121  
INITDIAG procedure 4-121  
INQUIRY (inquiry mode) condition 3-33  
inquiry mode (INQUIRY) Condition 3-33  
INQUIRY procedure 4-121  
INQUIRY# procedure 4-123  
instruction record 4-279  
intrasystem communications 5-93  
IPL procedure 4-124  
ITF procedure 4-124

IWLOAD procedure 4-125  
IWPTLOAD procedure 4-125  
IWPTSAVE procedure 4-125  
IWSAVE procedure 4-125

## J

job queue (JOBQ) condition 3-33  
job step 2-1  
JOBQ (job queue) condition 3-33  
JOBQ control command 6-12  
JOBQ OCL statement 5-52  
jobs 1-15  
JOBSTR procedure 4-125, A-51

## K

#KASRT utility program A-69  
KEYS procedure 4-128  
KEYSORT procedure 4-129

## L

\$LABEL utility program A-36  
LANLOAD procedure 4-129  
LANSAVE procedure 4-129  
length (?C'value') expression 3-13  
libraries  
    creating and maintaining 1-8  
    directory listing 4-151  
    list information A-52  
    status listing 4-151  
library  
    information 3-4  
    load members (LOAD) condition 3-34  
    members listing 4-154  
    members, reference numbers 1-18  
    procedure members (PROC) condition 3-36  
    source members (SOURCE) condition 3-37  
    subroutine members (SUBR) condition 3-38  
    unused space A-43  
LIBRARY OCL statement 5-54  
LIBRLIBR procedure 4-129, A-44  
licensed programs, managing 4-2  
line connection parameters 5-89  
line format record fields 4-274  
LINES procedure 4-132  
list  
    contents of exchange tape files A-63  
    information about libraries A-52  
    library members and information A-51  
LIST procedure 4-134  
LIST# procedure 4-135  
LISTDATA procedure 4-137  
LISTDATA/LISTFILE procedures A-11



**OLPDSAVE procedure 4-172****Operation Control Language (OCL) statements**

ABEND 5-5  
 ALLOCATE 5-6  
 ATTR 5-8  
 called by OS/400 procedures 4-1  
 CANCEL 5-13  
 CHANGE 5-14  
 COMM 5-16  
 comments 5-4  
 COMPILE 5-17  
 continuing 5-4  
 DATE 5-19  
 DEALLOC 5-21  
 DEBUG 5-22  
 EVOKE 5-23  
 FILE (for diskette files) 5-33  
 FILE (for tape files) 5-37  
 file information when writing A-68  
 FILE OCL (for disk files) 5-25  
 FORMS 5-46  
 general coding rules 5-3  
 identifiers 5-2  
 IMAGE 5-49  
 INCLUDE OCL 5-49  
 INFOMSG 5-51  
 introduction 1-2, 5-1  
 JOBQ 5-52  
 LOAD 1-2, 5-55  
 LOCAL 5-56  
 LOG 5-58  
 MEMBER 5-59  
 MENU 5-61  
 MSG 5-62  
 NOHALT 5-65  
 OFF 5-66  
 parameters 5-2  
 placement 5-1  
 POWER 5-67  
 PRINTER 5-68  
 procedure parameters 5-2  
 PROMPT 5-77  
 REGION 5-82  
 RESERVE 5-82  
 RUN 1-2, 5-83  
 RUN400 5-83  
 SESSION  
   APPC and APPN 5-85  
   asynchronous communications 5-91  
   BSCCEL and RPG II telecommunications 5-87  
   data record parameters 5-90  
   description 5-84  
   finance communications 5-91  
   intrasystem communications 5-93  
   line connection parameters 5-89  
   retail communications 5-92  
   SNA Upline Facility (SNUF) 5-86

**Operation Control Language (OCL) statements (continued)**

SETDEV 5-94  
 START 5-95  
 STOP 5-96  
 SWITCH 5-97  
 SYSLIST 5-98  
 TIMERSET 5-100  
 types of information 5-1  
 VARY 5-102  
 WAIT 5-103  
 WORKSTN 5-105

**operator's user ID (?USER?) expression 3-23****ORGANIZE procedure 4-172****OVERRIDE procedure 4-172****P****\$PACK utility program A-56****page format record fields 4-273****parameter (?n?) expression 3-8****parameter length (?Cn?) expression 3-12****parameters**

?n? expression 3-8  
 ?n'value'? default expression 3-9  
 attributes for USER 4-153  
 coding considerations 2-6  
 data record 5-90  
 defaults, procedure 2-5  
 forced value (?nF'value'?) expression 3-10  
 length (?Cn?) expression 3-12  
 line connection 5-89  
 maximum number 2-3  
 missing (?nR?) expression 3-11  
 missing message (?NR'mic'?) expression 3-12  
 OCL 5-2  
 positional 2-3  
 procedure 2-3, 5-2  
 required (?R?) expression 3-10  
 required (?R'mic'?) expression 3-11  
 substitution expressions 3-7  
 temporary value (?n'value'?) expression 3-9  
 testing 2-6

**parentheses 1-4****PASSTHRU procedure 4-173****PASSWORD procedure 4-174****password security (SECURITY) condition 3-36****PATCH procedure 4-175****PAUSE statement 3-53****PC**

See personal computer

**PCEXCH procedure 4-175****PCEXEC procedure 4-176****PCOLOAD procedure 4-176****PCOSAVE procedure 4-177**

**PCU procedure** 4-177  
**performance tips and coding techniques** 2-24  
**personal computer** 1-13  
**phone list completion (LISTDONE) condition** 3-34  
**\$PNLM utility program** A-56  
**positional parameters** 2-3  
**POST procedure** 4-178  
**\$POST utility program** A-57  
**POWER control command** 6-19  
**POWER OCL statement** 5-67  
**\$PRCED utility program** A-57  
**\$PRCLT utility program** A-57  
**print key information** A-59  
**PRINT procedure** 4-178  
**PRINTER OCL statement** 5-68  
**printers** 1-15  
**PRINTKEY procedure** 4-182, A-59  
**\$PRLST utility program** A-57  
**problem determination** 1-17  
**PROBLEM procedure** 4-185  
**PROC (library procedure members) condition** 3-36  
**procedure parameters** 5-2  
**procedures**  
    #STRTUP1 4-3  
    #STRTUP2 4-3  
    ALERT 4-4  
    allowed statements 2-1  
    ALOCFLDR 4-4  
    ALOCLIBR 4-5, A-42  
    ALTERBSC 4-5  
    ALTERCOM 4-5, A-59  
    ALTERSDL 4-10  
    APAR 4-10  
    APPNINFO 4-10  
    ARCHIVE 4-11  
    ASM 4-11  
    ASMLOAD 4-11  
    ASMSAVE 4-11  
    attributes 2-9  
    AUTO 4-11  
    AUTOC 4-11  
    BACKUP 4-14  
    BALPRINT 4-14  
    BASIC 4-14  
    BASICP 4-15  
    BASICR 4-15  
    BASICS 4-15  
    BASLOAD 4-15  
    BASSAVE 4-15  
    BGUATTR 4-16  
    BGUCHAR 4-16  
    BGUDATA 4-16  
    BGUGRAPH 4-16  
    BGULOAD 4-17  
    BGUSAVE 4-17  
    BLDFILE 4-17, A-33

**procedures (continued)**  
    BLDGRAPH 4-19  
    BLDINDEX 4-20, A-34  
    BLDLIBR 4-22, A-37  
    BLDMENU 4-25  
    BUILD 4-29  
    CACHE 4-29  
    calling from another procedure 2-8  
    CATALOG 4-29  
    CGU 4-39  
    CGULOAD 4-39  
    CGUSAVE 4-39  
    chart of OS/400 utilities and OCL statements  
      called 4-1  
    CHGX LATE 4-40  
    CHNGEMEM 4-40, A-43  
    CL commands 2-2, 2-20  
    CLRPF 4-43  
    CNFIGICF 4-43  
    CNFIGSSP 4-43  
    CNFIGX25 4-44  
    COBLOAD 4-44  
    COBOL 4-44  
    COBOLC 4-44  
    COBOLCG 4-47  
    COBOLG 4-47  
    COBOLONL 4-47  
    COBOLP 4-47  
    COBSAVE 4-48  
    COBSDA 4-48  
    COBSEU 4-48  
    commands 2-2, 2-20  
    comparing and evaluating values and branching 3-5  
    COMPRESS 4-49  
    CONDENSE 4-49, A-43  
    continuing statements 2-7  
    control expressions 2-2  
      canceling 3-6  
      comparing and evaluating values and  
      branching 3-5  
      data file information 3-4  
      diskette information 3-5  
      displaying messages or display formats 3-3  
      library information 3-4  
      substituting values and information 3-1  
      tape information 3-5  
      testing the procedure or job environment 3-6  
    COPYDATA 4-50, A-7  
    COPYDIAG 4-55  
    COPYI1 4-55  
    COPYPRT 4-57  
    CREATE 4-66  
    CSALL 4-71  
    DATE 4-71  
    debugging 2-27  
    DEFINEID 4-73

**procedures (continued)**

DEFINEPN 4-73  
definition 1-1, 2-1  
DEFINLOC 4-74  
DEFINX21 4-74  
DEFINX25 4-75  
DEFSUBD 4-75  
DELETE 4-76  
DELNRD 4-80  
description 4-1  
DFA 4-80  
DFU 4-80  
DFULOAD 4-81  
DFUSAVE 4-81  
DICTLOAD 4-81  
DICTSAVE 4-81  
directory 4-1  
DISABLE 4-81  
DISPLAY 4-82  
DLSLOAD 4-82  
DLSSAVE 4-82  
DOCCNV 4-82  
DOCPLOAD 4-82  
DOCPSAVE 4-82  
DSPLOCKS 4-83  
DSPMNU 4-83  
DSPSYS 4-83  
DSU 4-83  
DSULOAD 4-85  
DSUSAVE 4-85  
DUMP 4-86  
EDITNRD 4-86  
EM3270 4-86  
ENABLE 4-89  
ending 3-6  
ENTER 4-89  
ENTER# 4-91  
entering into the system 2-2  
EP3270 4-92  
EPLMRG 4-92  
ERR 4-93  
ES3270 4-95  
EVOKED condition 3-32  
examples 2-10  
EXTRACT 4-97  
FILELIB 5-43  
FLIB 4-98  
FORMAT 4-99  
FORTRAN 4-103  
FROMLIBR 4-103, A-45  
HELP 4-109  
HISTCOPY 4-114  
HISTCRT 4-114  
HISTORY 4-115  
ICFDEBUG 4-115  
ICVERIFY 4-115

**procedures (continued)**

IDDU 4-115  
IDDUDCT 4-116  
IDDUDFN 4-116  
IDDUDISK 4-116  
IDDULINK 4-116  
IDDUPRT 4-117  
IDDURBLD 4-118  
IDDUXLAT 4-119  
IGC 4-119  
INIT 4-119  
INIT9332 4-121  
INITDIAG 4-121  
INQUIRY 4-121  
INQUIRY# 4-123  
IPL 4-124  
ITF 4-124  
IWLOAD 4-125  
IWPTLOAD 4-125  
IWPTSAVE 4-125  
IWSAVE 4-125  
JOBSTR 4-125, A-51  
KEYS 4-128  
KEYSORT 4-129  
LANLOAD 4-129  
LANSAVE 4-129  
levels 2-8  
LIBRLIBR 4-129, A-44  
LINES 4-132  
LIST 4-134  
LIST# 4-135  
LISTDATA 4-137  
LISTDATA/LISTFILE A-11  
LISTFILE 4-142, A-3, A-52, A-63  
LISTKEYS 2-11  
LISTLIBR 4-147, A-51  
LISTNRD 4-154  
LOAD3601 4-155  
LOG 4-155  
LRTRLOAD 4-156  
LRTRSAVE 4-156  
MAINTX25 4-156  
making your own 2-1  
MCSCONV 4-156  
MOVEFLDR 4-156  
MRT 2-9  
MS3270 4-157  
MSDOWNL 4-157  
MSGFILE 4-158  
MSRJE 4-159  
naming 2-2  
nested 2-8  
NOHALT 4-161  
OCL statements 2-2  
OFCEPRT 4-162  
OFCCAL 4-163

**procedures** *(continued)*

OFCCANCL 4-163  
OFCCOMM 4-164  
OFCCONV 4-164  
OFCDATA 4-164  
OFCDFLT 4-165  
OFCDIR 4-165  
OFCFILE 4-166  
OFCGRP 4-166  
OFCINSTL 4-167  
OFCLDF 4-167  
OFCLOAD 4-167  
OFCMAIL 4-167  
OFCMAINT 4-169  
OFCMSG 4-169  
OFCQ 4-170  
OFCSAVE 4-170  
OFCSRCH 4-170  
OFCSTART 4-170  
OFCSTAT 4-171  
OFCUSER 4-171  
OLINK 4-172  
OLPDLOAD 4-172  
OLPDSAVE 4-172  
ORGANIZE 4-172  
OVERRIDE 4-172  
parameter defaults 2-5  
parameters 2-3  
PASSTHRU 4-173  
PASSWORD 4-174  
PATCH 4-175  
PCEXCH 4-175  
PCEXEC 4-176  
PCOLOAD 4-176  
PCOSAVE 4-177  
PCU 4-177  
PDOWNL 4-92  
performance tips and coding techniques 2-24  
POST 4-178  
PRINT 4-178  
PRINTKEY 4-182, A-59  
PROBLEM 4-185  
PROFLOAD 4-185  
PROFSAVE 4-185  
PRTGRAPH 4-186  
PTF 4-187  
QRY 4-187  
QRYDE 4-187  
QRYLOAD 4-188  
QRYRUN 4-188  
QRYSAVE 4-191  
READINFO 4-192  
REBLD 4-192  
RELOAD 4-192  
REMOVE 4-192, A-53  
RENAME 4-194

**procedures** *(continued)*

REQUESTX 4-195  
RESPONSE 4-196, A-3  
RESTEXTN 4-202  
RESTFLDR 4-203  
RETLIBR 4-203, A-55  
RESTNRD 4-205, A-14  
RESTORE 4-206, A-14  
RETRIEVE 4-212  
RGZFILE 4-212  
RJFILE 4-213  
RJTABLE 4-213  
ROLLKEYS 4-214  
RPG 4-214  
RPGC 4-214  
RPGLOAD 4-216  
RPGONL 4-217  
RPGP 4-217  
RPGR 4-217  
RPGSAVE 4-218  
RPGSDA 4-218  
RPGSEU 4-219  
RPGX 4-219  
SAMPLE 2-10  
SAVE 4-220, A-19  
SAVEEXTN 4-227  
SAVEFLDR 4-228  
SAVELIBR 4-228, A-54  
SAVENRD 4-230, A-24  
SCRNPRT 2-14  
SDA 4-230  
SDALOAD 4-231  
SDASAVE 4-232  
SECDEF 4-232  
SECEDIT 4-232  
SECLIST 4-233  
SECRET 4-234  
SECSAVE 4-234  
SERVICE 4-234  
SERVLOG 4-235  
SET 4-235, A-59  
SETALERT 4-237, A-3  
SETCOMM 4-237  
SETDUMP 4-242  
SEU 4-242  
SEULOAD 4-243  
SEUSAVE 4-243  
SHRFLOAD 4-244  
SHRFSAVE 4-244  
SLIB 4-244  
SMF 4-245  
SMFDATA 4-245  
SMFPRINT 4-245  
SMFSTART 4-246  
SMFSTOP 4-246  
SOFTWARE 4-246

**procedures** *(continued)*

SORT 4-247  
SPECIFY 4-247  
SRTX 4-248  
SRTXBLD 4-249  
SRTXLOAD 4-249  
SRTXSAVE 4-249  
starting automatically 1-17  
STARTM 4-249  
STATEST 4-250  
STOPGRP 4-251  
STOPM 4-251  
STRTGRP 4-251  
SWDLOAD 4-252  
SWDSAVE 4-252  
SWITCH 4-252  
SYSLIST 4-253  
SYSTYPE 4-255  
TAPECOPY 4-255, A-63  
TAPEINIT 4-262  
TAPESTAT 4-263  
testing the job environment 3-6  
TEXTCONV 4-263  
TEXTDCT 4-264  
TEXTDOC 4-264  
TEXTFLDR 4-280  
TEXTLOAD 4-280  
TEXTOBJ 4-281  
TEXTPROF 4-281  
TEXTPRTQ 4-281  
TEXTREL 4-281  
TEXTSAVE 4-282  
TIMER 4-282  
TOLIBR 4-283, A-50  
TRACE 4-288  
TRANSFER 4-289, A-4  
TRNMGR 4-293  
troubleshooting 2-27  
UPDATE 4-294  
UPDATE# 4-295  
utility control statements 2-2  
WRKSPL 4-297  
WRKUSER 4-297  
WSFLOAD 4-297  
WSFSAVE 4-297  
WSU 4-298  
XREST 4-298  
XSAVE 4-298

**processing**  
diskettes 1-9  
tapes 1-10

**PROFLOAD procedure** 4-185  
**PROFSAVE procedure** 4-185  
**program date (?DATE?) expression** 3-16  
**programming considerations** 1-3

**programs** 1-11  
**programs and procedures, running** 1-13  
**PROMPT OCL statement** 5-77  
**\$PRPWD utility program** A-57  
**PRTFILE files** 4-268  
**PRTGRAPH procedure** 4-186  
**PRTY control command** 6-19  
**\$PRUED utility program** A-57  
**\$PRUID utility program** A-57  
**\$PRURS utility program** A-57  
**\$PRUSV utility program** A-58  
**PTF procedure** 4-187

**Q**

**QEXCVTDV program** E-1  
**QEXRCDBK program** E-2  
**QEXRMVDE program** E-2  
**QEXSHRO program** E-4  
**QRY procedure** 4-187  
**QRYDE procedure** 4-187  
**QRYLOAD procedure** 4-188  
**QRYRUN procedure** 4-188  
**QRYSAVE procedure** 4-191

**R**

**READINFO procedure** 4-192  
**REBLD procedure** 4-192  
**record-mode file, creating** A-46  
**REGION OCL statement** 5-82  
**related printed information** H-1  
**RELEASE control command** 6-19  
**RELOAD procedure** 4-192  
**remote job entry** 4-159, 4-213  
**remove members from library** A-53  
**REMOVE procedure** 4-192, A-53  
**\$RENAM utility program** A-58  
**RENAME procedure** 4-194  
**REPLY control command** 6-20  
**REQUESTX procedure** 4-195  
**required parameter (?R?) expression** 3-10  
**required parameter message (?R'mic'?)  
expression** 3-11  
**RESERVE OCL statement** 5-82  
**RESET statement** 3-54  
**RESPONSE procedure** 4-196, A-3  
**RESTART control command** 6-20  
**RESTEXTN procedure** 4-202  
**RESTFLDR procedure** 4-203  
**RETLIBR procedure** 4-203, A-55  
**RESTNRD procedure** 4-205, A-14  
**restore a library** A-55  
**RESTORE procedure** 4-206, A-14  
**restoring files** A-14

restoring the network resource directory A-14  
 retail communications 5-92  
 RETRIEVE procedure 4-212  
 return code 3-51  
 return code (?CD?) expression 3-13  
 return code definitions 3-13  
 RETURN statement 3-55  
 RGZFILE procedure 4-212  
 RJFILE procedure 4-213  
 RJTABLE procedure 4-213  
 ROLLKEYS procedure 4-214  
 RPG procedure 4-214  
 RPGC procedure 4-214  
 RPGLOAD procedure 4-216  
 RPGONL procedure 4-217  
 RPGP procedure 4-217  
 RPGR procedure 4-217  
 RPGSAVE procedure 4-218  
 RPGSDA procedure 4-218  
 RPGSEU procedure 4-219  
 RPGX procedure 4-219  
 \$RREDT utility program A-58  
 \$RRESC utility program A-58  
 \$RRLST utility program A-58  
 \$RRSAV utility program A-59  
 \$RRSTR utility program A-59  
 \$RRTED utility program A-59  
 \$RRTL utility program A-59  
 RUN OCL statement 1-2, 5-83  
 RUN400 OCL statement 5-83  
 running procedures (ACTIVE) condition 3-26

## S

SAMPLE procedure 2-10  
 save a library A-54  
 SAVE procedure 4-220, A-19  
 SAVEEXTN procedure 4-227  
 SAVEFLDR procedure 4-228  
 SAVELIBR procedure 4-228, A-54  
 SAVENRD procedure 4-230, A-24  
 saving files A-19  
 SCRNPRT procedure 2-14  
 SDA procedure 4-230  
 SDALOAD procedure 4-231  
 SDASAVE procedure 4-232  
 SECDEF procedure 4-232  
 SECEDIT procedure 4-232  
 SECLIST procedure 4-233  
 SECREST procedure 4-234  
 SECSAVE procedure 4-234  
 sector-mode file, creating A-45  
 SECURITY (password security) condition 3-36  
 SERVICE procedure 4-234  
 SERVLOG procedure 4-235

session files library (?SFLIB?) expression 2-21, 3-22  
 session library (?SLIB?) expression 3-22  
**SESSION OCL statement**  
   APPC and APPN 5-85  
   asynchronous communications 5-91  
   BSCEL and RPG II telecommunications 5-87  
   data record parameters 5-90  
   description 5-84  
   finance communications 5-91  
   intrasystem communications 5-93  
   line connection parameters 5-89  
   retail communications 5-92  
   SNA Upline Facility (SNUF) 5-86  
 session printer (?PRINTER?) expression 3-21  
 set display station environment A-59  
 SET procedure 4-235, A-59  
 SETALERT procedure 4-237, A-3  
 \$SETCF utility program A-59  
 SETCOMM procedure 4-237  
 \$SETCP utility program A-60  
 SETDEV OCL statement 5-94  
 SETDUMP procedure 4-242  
 setting the return code 3-51  
 SEULOAD procedure 4-243  
 SEUSAVE procedure 4-243  
 severity level, automatic response 5-65  
 SFGR system messages D-1  
 \$SFGR utility program A-60  
 SHRFLOAD procedure 4-244  
 SHRFSAVE procedure 4-244  
 \$SINCT utility program A-62  
 \$SINDL utility program A-62  
 \$SINR utility program A-62  
 SLIB procedure 4-244  
 SMF procedure 4-245  
 SMFDATA procedure 4-245  
 SMFPRINT procedure 4-245  
 SMFSTART procedure 4-246  
 SMFSTOP procedure 4-246  
 SNA Upline Facility (SNUF) 5-86  
 SNUF  
   See SNA Upline Facility  
 SOFTWARE procedure 4-246  
 SORT procedure 4-247  
 SOURCE (library source members) condition 3-37  
 source entry utility (SEU) procedure 4-242  
 special characters in names 2-22  
 specify print key information A-59  
 SPECIFY procedure 4-247  
 SRTX procedure 4-248  
 SRTXBLD procedure 4-249  
 SRTXLOAD procedure 4-249  
 SRTXSAVE procedure 4-249  
 start a job A-51

**START control command** 6-22  
**start document record fields** 4-268  
**start error log record fields** 4-279  
**START OCL statement** 5-95  
**STARTM procedure** 4-249  
**statement portion of IF conditional expression** 3-43  
**statements**  
// \* (informational message) 3-44  
// \*\* (system console message) 3-45  
calculating 3-48  
CANCEL 3-46  
CEND A-47  
comment (\*) 3-7, 4-69, 4-200  
COPY A-47  
end of data (/) 5-107  
EVALUATE 3-46  
FILE A-68  
GOTO 3-52  
IF conditional expression 3-43  
message control 4-67  
message member 4-67  
message text 4-68  
OCL 2-2  
PAUSE 3-53  
RESET 3-54  
RETURN 3-55  
System/34 compatible A-24  
TAG 3-52  
**STATEST procedure** 4-250  
**STATUS control command** 6-24  
**status listing** 4-151  
**STATUSF control command** 6-30  
**STOP control command** 6-32  
**STOP OCL statement** 5-96  
**STOPGRP procedure** 4-251  
**STOPM procedure** 4-251  
**string1=string2 (comparing, equal to)**  
condition 3-40  
**string1>string2 (comparing, greater than)**  
condition 3-41  
**STRTPGRP procedure** 4-251  
**#STRTP1 procedure** 4-3  
**#STRTP2 procedure** 4-3  
**SUBR (library subroutine members) condition** 3-38  
**substituting values and information** 3-1  
**substitution expressions**  
?n? 2-3  
?n'value'? 2-3  
?nF'value'? 2-4  
?nR? 2-4  
?nR'mic'? 2-4  
?nT'value'? 2-4  
?VOLID? or ?VOLID'location'? (diskette or tape  
volume ID) 3-23  
actual file size (?F'A,name'? or  
?F'A,name,date'?) 3-18

**substitution expressions (continued)**  
CL commands 2-20  
concatenate values 2-26  
current library (?CLIB?) 3-16  
current menu (?MENU?) 3-21  
date format (?DATEFMT?) 3-17  
default parameter (?n'value'?) 3-9  
definition 3-7  
device name (?DEV'unit'?) 3-17  
display station ID (?WS?) 3-24  
evaluating 3-51  
file size (?F'S,name'? or ?F'S,name,date'?) 3-17  
files library (?FLIB?) 3-19  
files type (?F'T,name'?) 3-19  
first level procedure (?PROC?) 3-22  
forced value parameter (?nF'value'?) 3-10  
length (?C'value'?) 3-13  
local data area (?L'position,length'?) 3-19  
message ID (?MSGID?) 3-21  
message member (?Mmic? or  
?M'mic,position,length'?) 3-20  
missing parameter (?nR?) 3-11  
missing parameter message (?NR'mic'?) 3-12  
nested 3-24  
operator's user ID (?USER?) 3-23  
parameter (?n?) 3-8  
parameter length (?Cn?) 3-12  
program date (?DATE?) 3-16  
required parameter (?R?) 3-10  
required parameter (?R'mic'?) 3-11  
restrictions 3-7  
return code (?CD?) 3-13  
session files library (?SFLIB?) 3-22  
session library (?SLIB?) 3-22  
session printer (?PRINTER?) 3-21  
system list device (SYSLIST?) 3-22  
system time (?TIME?) 3-23  
temporary value (?n'value'?) 3-9  
**\$SVCASR utility program** A-62  
**SWDLOAD procedure** 4-252  
**SWDSAVE procedure** 4-252  
**SWITCH (switches) condition** 3-39  
**SWITCH OCL statement** 5-97  
**SWITCH procedure** 4-252  
**switch settings** 4-252  
**switches (SWITCH) condition** 3-39  
**SWITCHn (individual switches) condition** 3-39  
**syntax diagram** 1-5  
**syntax formats**  
conventions used for describing  
braces 1-3  
brackets 1-4  
capitalized expressions 1-3  
commas 1-5  
parentheses 1-4  
underscores 1-4

**SYSLIST OCL statement** 5-98  
**SYSLIST procedure** 4-253  
**system communication** 1-17  
**system console (CONSOLE) condition** 3-27  
**system console message (// \*\*) statement** 3-45  
**system list device (SYSLIST?) expression** 3-22  
**system maintenance** 1-17  
**system messages, SFGR** D-1  
**system support** 1-6  
**system time (?TIME?) expression** 3-23  
**SYSTYPE procedure** 4-255

## T

**TAG and GOTO statements** 3-52  
**tape**  
   information 3-5  
   label listings 4-39  
   processing 1-10  
**TAPECOPY procedure** 4-255, A-63  
**TAPEINIT procedure** 4-262  
**TAPESTAT procedure** 4-263  
**\$TCOPY utility program** A-62  
**temporary value parameter (?n'value?)  
 expression** 3-9  
**text**  
   base line record fields 4-277  
   half index down record fields 4-278  
   half index up record fields 4-276  
   lines 4-276  
**TEXTCONV procedure** 4-263  
**TEXTDCT procedure** 4-264  
**TEXTDOC procedure** 4-264  
**TEXTFLDR procedure** 4-280  
**TEXTLOAD procedure** 4-280  
**TEXTOBJ procedure** 4-281  
**TEXTPROF procedure** 4-281  
**TEXTPRTQ procedure** 4-281  
**TEXTREL procedure** 4-281  
**TEXTSAVE procedure** 4-282  
**TIME control command** 6-34  
**TIMER procedure** 4-282  
**TIMERSET OCL statement** 5-100  
**\$TINIT utility program** A-64  
**\$TMSERV utility program** A-64  
**TOLIBR procedure** 4-283, A-50  
**TRACE procedure** 4-288  
**TRANSFER procedure** 4-289, A-4  
**TRNMGR procedure** 4-293  
**troubleshooting procedures** 2-27

## U

**\$UASC utility program** A-65  
**\$UASF utility program** A-65

**underscores** 1-4  
**unused library space** A-43  
**UPDATE procedure** 4-294  
**UPDATE# procedures** 4-295  
**\$DUPRD utility program** A-32  
**utilities**  
   \$IEDS A-35  
   called by OS/400 procedures 4-1  
   making you own procedures A-1  
**utility control statements**  
   COPY 1-2  
   definition 1-2  
   END 1-2  
**utility programs**  
   \$ARSP A-3  
   \$BICR A-3  
   \$BMENU A-5  
   \$BUILD A-6  
   \$COPY A-7  
   \$CPPE A-28  
   \$CZUT A-28  
   \$DCOPY A-29  
   \$DDST A-29  
   \$DELET A-29  
   \$DPGP A-31  
   \$DPGR A-32  
   \$DUPRD A-32  
   \$FBLD A-33  
   \$FREE A-34  
   \$HELP A-35  
   \$HIST A-35  
   \$IDSET A-35  
   \$INIT A-35  
   \$LABEL A-36  
   \$MAINT A-37  
   \$MGBLD A-55  
   \$MMST A-56  
   \$PACK A-56  
   \$PNLM A-56  
   \$POST A-57  
   \$PRCED A-57  
   \$PRCLT A-57  
   \$PRLST A-57  
   \$PRPWD A-57  
   \$PRUED A-57  
   \$PRUID A-57  
   \$PRURS A-57  
   \$PRUSV A-58  
   \$RENAM A-58  
   \$RREDT A-58  
   \$RRESC A-58  
   \$RRLST A-58  
   \$RRSAV A-59  
   \$RRSTR A-59  
   \$RRTED A-59  
   \$RRTLT A-59

**utility programs** *(continued)*

\$SETCF A-59  
\$SETCP A-60  
\$SFGR A-60  
\$SINCT A-62  
\$SINDL A-62  
\$SINR A-62  
\$SVCASR A-62  
\$TCOPY A-62  
\$TINIT A-64  
\$TMSERV A-64  
\$UASC A-65  
\$UASF A-65  
\$XNLM A-66  
\$XNSH A-66  
\$XREST A-66  
\$XSAVE A-66  
#GCFR A-66  
#GSORT A-66  
#IENBL A-35  
#KASRT A-69

## **V**

VARY control command 6-34  
VARY OCL statement 5-102  
VOLID (diskette and tape volume IDs)  
condition 3-41

## **W**

WAIT OCL statement 5-103  
WORKSTN OCL statement 5-105  
writing OCL statements A-68  
WRKSPL procedure 4-297  
WRKUSER procedure 4-297  
WSFLOAD procedure 4-297  
WSFSAVE procedure 4-297  
WSU procedure 4-298

## **X**

\$XNLM utility program A-66  
\$XNSH utility program A-66  
XREST procedure 4-298  
\$XREST utility program A-66  
XSAVE procedure 4-298  
\$XSAVE utility program A-66

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

AS/400 Advanced Series  
System/36 Environment Reference  
Version 3

Publication No. SC41-4731-00

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

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

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				

T H A N K   Y O U !

Please tell us how we can improve this manual:

---



---



---



---



---

May we contact you to discuss your responses?  Yes  No

Phone: (\_\_\_\_) \_\_\_\_\_ Fax: (\_\_\_\_) \_\_\_\_\_ Internet: \_\_\_\_\_

**To return this form:**

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

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

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

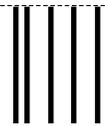
\_\_\_\_\_  
Phone No.



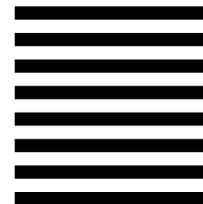
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



# BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

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



Fold and Tape

Please do not staple

Fold and Tape





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

SC41-4731-00





**AS/400 Advanced Series**

**System/36 Environment Reference**

**Version 3**