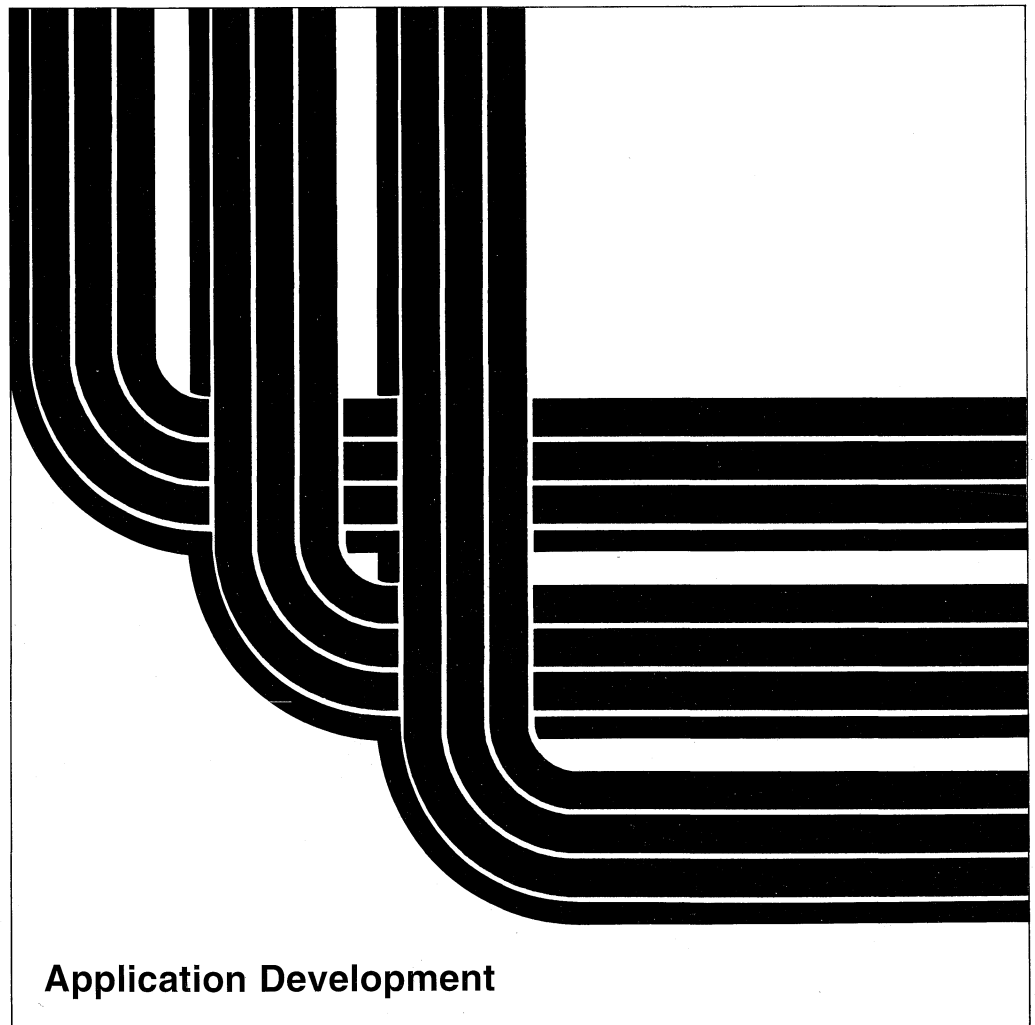


**Office Services Concepts and
Programmer's Guide**

Version 2





Application System/400

SC41-9758-02

**Office Services Concepts and
Programmer's Guide**

Version 2

Take Note!

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

Third Edition (November 1993)

This edition applies to the licensed programs IBM Operating System/400 (Program 5738-SS1), Version 2 Release 3 Modification 0, and OfficeVision/400 (Program 5738-WP1), Version 2 Release 3 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. Publications are not stocked at the address given below.

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

IBM Corporation
Information Development
Attn. Department 5D9
5 West Kirkwood Boulevard
Roanoke, TX 76299-0001

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 1991, 1993. All rights reserved.

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

Contents

| | |
|---|------|
| Notices | xiii |
| Programming Interface Information | xiii |
| Trademarks and Service Marks | xiv |
| | |
| About This Guide | xv |
| Who Should Use This Guide | xv |
| Programming Interface Information | xv |
| | |
| Summary of Changes | xvii |
| | |
| Chapter 1. Office Command Interfaces | 1-1 |
| Relationships between Services and Products | 1-2 |
| Common Considerations for Service Administration | 1-3 |
| Office Services Command Interfaces | 1-4 |
| Calendar Services | 1-6 |
| Directory Services | 1-6 |
| Document Distribution Services | 1-9 |
| Document Library Services | 1-9 |
| Security Services | 1-10 |
| Text Search Services | 1-11 |
| Word Processing Services | 1-13 |
| Miscellaneous Services | 1-14 |
| Integrating Applications with OfficeVision/400 Using Commands | 1-16 |
| Customizing the Main Menu | 1-16 |
| Using OfficeVision/400 Functions with User Application Menus | 1-16 |
| Using Office Services with User Applications | 1-18 |
| | |
| Chapter 2. Calendar Services | 2-1 |
| Calendar Services Overview | 2-1 |
| Calendar Services Objects | 2-2 |
| Common CL Command Parameter Information | 2-2 |
| Library Parameters | 2-2 |
| Date Parameters | 2-3 |
| Time Parameters | 2-3 |
| ITMID Parameters | 2-3 |
| CAL Parameter | 2-3 |
| GRPCALS Parameter | 2-3 |
| Authority Parameters | 2-4 |
| Character Set and Code Page Considerations | 2-4 |
| CMDCHRID Parameter | 2-4 |
| Other Calendar Services Topics | 2-5 |
| Security Considerations | 2-5 |
| Calendar Services Control Language Commands | 2-5 |
| Add Calendar Item (ADDCALITM) Command | 2-5 |
| Change Calendar Authority (CHGCALAUT) Command | 2-7 |
| Create Calendar (CRTCAL) Command | 2-8 |
| Delete Calendar (DLTCAL) Command | 2-10 |
| Display Calendar Authority (DSPCALAUT) Command | 2-11 |
| Display Calendar Description (DSPCALD) Command | 2-11 |
| Display Calendar Item (DSPCALITM) Command | 2-12 |

| | |
|--|------------|
| End Calendar Service (ENDCALSRV) Command | 2-12 |
| Print Calendar (PRTCAL) Command | 2-12 |
| Query Calendar Item (QRYCALITM) Command | 2-13 |
| Remove Calendar Item (RMVCALITM) Command | 2-15 |
| Restore Calendar (RSTCAL) Command | 2-16 |
| Save Calendar (SAVCAL) Command | 2-17 |
| Start Calendar Service (STRCALSRV) Command | 2-18 |
| Calendar User-Defined Functions | 2-19 |
| Security | 2-19 |
| Function Names | 2-20 |
| Substitution Variables | 2-20 |
| Examples | 2-24 |
| Control Calendar API (QOOCTLCL) | 2-28 |
| Control Calendar API Example | 2-30 |
| Chapter 3. Directory Services | 3-1 |
| Directory Services Overview | 3-1 |
| System Distribution Directory | 3-2 |
| Distribution Lists | 3-2 |
| Directory Services Objects | 3-3 |
| Directory Files | 3-3 |
| Distribution Recipient Index | 3-5 |
| Distribution List Files | 3-5 |
| Directory Services Commands | 3-6 |
| Other Directory Services Topics | 3-7 |
| Security Considerations | 3-7 |
| Directory Services Control Language Interfaces | 3-10 |
| Add Directory Entry (ADDDIRE) Command | 3-10 |
| Change Directory Attributes (CHGDIRA) Command | 3-12 |
| Change Directory Entry (CHGDIRE) Command | 3-12 |
| Display Directory (DSPDIR) Command | 3-13 |
| Remove Directory Entry (RMVDIRE) Command | 3-15 |
| Rename Directory Entry (RNMDIRE) Command | 3-16 |
| Work with Directory (WRKDIR) Command | 3-17 |
| Distribution List Commands | 3-19 |
| Add Distribution List Entry (ADDDSTLE) Command | 3-19 |
| Create Distribution List (CRTDSTL) Command | 3-20 |
| Delete Distribution List (DLTDSTL) Command | 3-22 |
| Display Distribution List (DSPDSTL) Command | 3-23 |
| Remove Distribution List Entry (RMVDSTLE) Command | 3-24 |
| Work with Distribution List (WRKDSTL) Command | 3-25 |
| Example Applications | 3-26 |
| Maintaining Multiple Distribution Lists | 3-26 |
| Updating the Directory from Customer Data | 3-32 |
| Chapter 4. Document Distribution Services | 4-1 |
| Document Distribution Services Overview | 4-1 |
| Using Document Distribution Services to Send, Receive, and Manage Mail | 4-1 |
| Mail Logs | 4-2 |
| Relationship to SNADS | 4-3 |
| VM/MVS Bridge | 4-3 |
| OSI Message Services/400 | 4-3 |
| Simple Mail Transfer Protocol | 4-4 |
| Document Distribution Services Objects | 4-5 |

| | |
|--|------------|
| Distribution Recipient Index | 4-6 |
| Distribution Tracking Object | 4-6 |
| Document Unit Object | 4-6 |
| Dumping Internal Mail Logs | 4-7 |
| Document Distribution Services Application Interfaces | 4-9 |
| Other Document Distribution Services Topics | 4-10 |
| Damaged Objects | 4-10 |
| Document Distribution Services Jobs in the QSNADS Subsystem | 4-10 |
| Indirect User Data Queue Interface | 4-13 |
| Security Considerations (Working on Behalf of Another User) | 4-17 |
| Document Distribution Services Control Language Interfaces | 4-17 |
| Change Distribution Description (CHGDSTD) Command | 4-18 |
| Delete Distribution (DLTDST) Command | 4-19 |
| Query Distribution (QRYDST) Command | 4-20 |
| Receive Distribution (RCVDST) Command | 4-22 |
| Send Document (SNDDOC) Command | 4-25 |
| Send Distribution (SNDDST) Command | 4-26 |
| Example Applications | 4-29 |
| Granting User Permission to All User Profiles in the Directory | 4-29 |
| Sending Messages, Notes, and Documents to Office Users | 4-33 |
| Receiving Messages, Notes, and Documents from a User's Mailbox | 4-36 |
| Delegating Mail | 4-40 |
| Writing Programs to Query a User's Mailbox | 4-60 |
| Converting a Spooled File to FFTDCA and Distributing the File | 4-61 |
| Maintaining Mail Logs | 4-73 |
| Building a Bridge between Office and Object Distribution | 4-88 |
| Identifying New Mail | 4-95 |
| Deleting Outgoing Mail | 4-96 |
| | |
| Chapter 5. Document Library Services | 5-1 |
| Document Library Services Overview | 5-1 |
| Accessing Document Library Services | 5-2 |
| Host Display Station Interface | 5-3 |
| Remotely Attached System Interface | 5-4 |
| Locally Attached PC Interface | 5-4 |
| Document Library Services Objects | 5-5 |
| Documents | 5-5 |
| Folders and Document Lists | 5-6 |
| PC Files and Folders | 5-7 |
| Object Ownership | 5-8 |
| Document Library Object Naming | 5-8 |
| Other Document Library Services Topics | 5-9 |
| Enrollment Requirements | 5-9 |
| Document Library Security Considerations | 5-9 |
| Document List Considerations | 5-10 |
| Access Codes | 5-10 |
| Considerations When Working on Behalf of Another User | 5-10 |
| Folder Security on the Personal Computer | 5-10 |
| Character Set and Code Page Considerations | 5-11 |
| Work in Process Flag or Allow Replacement Value | 5-11 |
| Checking Out and Checking In Documents | 5-11 |
| Document Library Services Control Language Interfaces | 5-12 |
| Change Document Description (CHGDOCD) Command | 5-14 |
| Check Document Library Object (CHKDLO) Command | 5-15 |

| | |
|---|------|
| Copy Document (CPYDOC) Command | 5-17 |
| Copy From PC Document (CPYFRMPCD) Command | 5-20 |
| Copy To PC Document (CPYTOPCD) Command | 5-21 |
| Create Folder (CRTFLR) Command | 5-23 |
| Delete Document Library Object (DLTDLO) Command | 5-24 |
| Display Document Library Object Name (DSPDLONAM) Command | 5-26 |
| Display Folder (DSPFLR) Command | 5-27 |
| File Document (FILDOC) Command | 5-29 |
| Move Document (MOVDOC) Command | 5-33 |
| Query Document Library (QRYDOCLIB) Command | 5-36 |
| Rename Document Library Object (RNMDLO) Command | 5-38 |
| Replace Document (RPLDOC) Command | 5-39 |
| Retrieve Document Library Object Name (RTVDLONAM) Command | 5-40 |
| Retrieve Document (RTVDOC) Command | 5-42 |
| Example Applications | 5-45 |
| | |
| Chapter 6. Security Services | 6-1 |
| Security Services Overview | 6-1 |
| Access Control Functions | 6-2 |
| AS/400 Enrollment | 6-3 |
| User Profiles and Group Profiles | 6-3 |
| Specific User Authorization | 6-3 |
| Group Authorization | 6-3 |
| Ownership Rights | 6-4 |
| Public Authorization | 6-4 |
| Special Authority | 6-5 |
| Authorization Lists | 6-7 |
| Working on Behalf of Another User | 6-9 |
| Security Services for Document Library Objects | 6-12 |
| Document and Folder Auditing | 6-12 |
| Document and Folder Security | 6-12 |
| Document and Folder Authority Levels | 6-13 |
| Access Codes | 6-17 |
| Security Services Control Language Interfaces | 6-19 |
| Add Document Library Object Authority (ADDDLOAUT) Command | 6-19 |
| Change Document Library Object Auditing Level (CHGDLOAUD) Command | 6-20 |
| Change Document Library Object Authority (CHGDLOAUT) Command | 6-20 |
| Change Document Library Object Owner (CHGDLOOWN) Command | 6-21 |
| Display Authorization List Document Library Objects (DSPAUTLDLO) Command | 6-22 |
| Display Document Library Object Auditing Level (DSPDLOAUD) Command | 6-23 |
| Display Document Library Object Authority (DSPDLOAUT) Command | 6-23 |
| Display User Permission (DSPUSRPMN) Command | 6-24 |
| Edit Document Library Object Authority (EDTDLOAUT) Command | 6-25 |
| Grant User Permission (GRTUSRPMN) Command | 6-26 |
| Remove Document Library Object Authority (RMVDLOAUT) Command | 6-26 |
| Revoke User Permission (RVKUSRPMN) Command | 6-27 |
| Example Applications | 6-28 |
| | |
| Chapter 7. Text Search Services | 7-1 |
| Text Search Services Overview | 7-1 |
| Text Search | 7-1 |
| Text Index | 7-1 |

| | |
|---|------|
| Scheduling Queue | 7-1 |
| Administration | 7-2 |
| Language Dictionaries | 7-2 |
| Linguistic Processing | 7-2 |
| Search Functions | 7-3 |
| Text Search Services Objects | 7-3 |
| Text Index | 7-3 |
| Scheduling Queue | 7-4 |
| Automatic Administration | 7-5 |
| Text Search Services Files | 7-5 |
| Language Dictionaries | 7-7 |
| Linguistic Processing | 7-8 |
| Language IDs | 7-12 |
| Stop Word Lists | 7-13 |
| Text Search Services CL Commands | 7-15 |
| Add Text Index Entry (ADDTXTIDX) Command | 7-15 |
| Display Text Index Status (DSPIDXSTS) Command | 7-16 |
| End Index Monitor (ENDIDXMON) Command | 7-16 |
| Remove Text Index Entry (RMVTEXTIDX) Command | 7-17 |
| Start Index Monitor (STRIDXMON) Command | 7-18 |
| Start Reorganization of Index (STRRGZIDX) Command | 7-18 |
| Start Update of Index (STRUPDIDX) Command | 7-18 |
| Work with Text Index (WRKTEXTIDX) Command | 7-19 |
| Change Document Language (ZCHGDOCLNG) Command Example | 7-19 |
| Index Documents (ZIDXDOC) Command Example | 7-24 |
| | |
| Chapter 8. Word Processing Services | 8-1 |
| Word Processing Services Overview | 8-1 |
| Document Functions | 8-1 |
| Printing and Merging Text and Data | 8-1 |
| Dictionaries | 8-2 |
| Text Profiles | 8-2 |
| Word Processing Services Objects | 8-2 |
| Text Profiles | 8-2 |
| Dictionaries | 8-3 |
| Database Files | 8-4 |
| Notepads | 8-5 |
| Help Text Documents | 8-5 |
| Queries | 8-5 |
| Object Naming Conventions | 8-6 |
| Word Processing Services Command Interfaces | 8-6 |
| Control Language Commands | 8-6 |
| Other Word Processing Services Topics | 8-7 |
| Editor Selection | 8-7 |
| Text Recovery | 8-8 |
| Document Interchange | 8-8 |
| Designing Online Help with Help Text Documents | 8-9 |
| Using the Run Instruction | 8-13 |
| Security Considerations | 8-14 |
| Word Processing Services Control Language Interfaces | 8-14 |
| Check Document (CHKDOC) Command | 8-14 |
| Create Document (CRTDOC) Command | 8-16 |
| Create Spelling Aid Dictionary (CRTSPADCT) Command | 8-17 |
| Delete Spelling Aid Dictionary (DLTSPADCT) Command | 8-20 |

| | |
|---|------|
| Display Document (DSPDOC) Command | 8-22 |
| Display Help Document (DSPHLPDOC) Command | 8-23 |
| Edit Document (EDTDOC) Command | 8-26 |
| Fill Form Document (FILLFORM) Command | 8-27 |
| Merge Document (MRGDOC) Command | 8-29 |
| Paginate Document (PAGDOC) Command | 8-32 |
| Print Document (PRTDOC) Command | 8-33 |
| Print Stop Word List (PRTSWL) Command | 8-38 |
| Retrieve Stop Word List Source (RTVSWLSRC) Command | 8-38 |
| Start Word Processing (STRWP) Command | 8-38 |
| Work with Documents (WRKDOC) Command | 8-39 |
| Work with Document Print Queue (WRKDOCPRQ) Command | 8-41 |
| Work with Folders (WRKFLR) Command | 8-42 |
| Work with Text Profiles (WRKTXTPRF) Command | 8-43 |
| Example Applications | 8-45 |
| Creating Fill Form Document | 8-45 |
| Printing All Documents in a Folder | 8-54 |
| Spell Checking All Changed Documents in a Folder | 8-56 |
| Saving Final Form Documents in a Folder | 8-57 |
| Printing Repetitive Letters | 8-60 |
| Creating Customized Billings | 8-60 |
| A Sample Customer Response Application | 8-62 |
| Chapter 9. Data Backup, Recovery, and Storage Management | 9-1 |
| Individual Object Restore | 9-3 |
| Restore of Individual Documents and Folders | 9-3 |
| Restore of Database Files | 9-4 |
| Restore of Directory | 9-4 |
| Restore of Calendars | 9-4 |
| Reclaim Document Library Object (RCLDLO) Command | 9-5 |
| Reorganize Document Library Object (RGZDLO) Command | 9-6 |
| Office Data Storage and Journal Management | 9-8 |
| Managing Storage Used by Documents and Folders | 9-8 |
| Managing Storage Used by Database Objects and Journals | 9-8 |
| Managing Storage Used by Distribution Objects | 9-11 |
| Office Object Individual Recovery | 9-11 |
| Document | 9-11 |
| Folders | 9-12 |
| Internal Document Library Objects | 9-12 |
| Logically Damaged Documents or Folders | 9-12 |
| Damaged Database Files | 9-13 |
| Damaged QUSRSYS or QDOC System Library | 9-15 |
| QDOC and QUSRSYS Library Mismatches | 9-15 |
| Chapter 10. Miscellaneous Services | 10-1 |
| Miscellaneous Services Overview | 10-1 |
| Configure VM/MVS Bridge (CFGRPDS) Command | 10-1 |
| Start Office (STROFC) Command | 10-2 |
| Change Office Attributes (CHGOFCA) Command | 10-3 |
| Retrieve Office Attributes (RTVOFCA) Command | 10-3 |
| Delete Outgoing Mail (DLTOUTMAIL) Command | 10-3 |
| Remote Document Library Processing | 10-4 |
| Accessing a Remote Library from an AS/400 System | 10-4 |
| Allowing Remote Users to Access an AS/400 System | 10-6 |

| | |
|--|-------|
| Work with Document Library (WRKDOCLIB) Command | 10-6 |
| Enrollment Commands | 10-7 |
| Add Office Enrollment (ADDOFCENR) Command | 10-7 |
| Change Office Enrollment (CHGOFCENR) Command | 10-8 |
| Remove Office Enrollment (RMVOFCENR) Command | 10-9 |
| Retrieve Office Enrollment (RTVOFCENR) Command | 10-10 |
| Example Applications | 10-10 |
| STROFC Examples | 10-10 |
| Initial Program to Call STROFC or Another User Menu | 10-13 |
| | |
| Chapter 11. Enhancing OfficeVision/400 | 11-1 |
| Operating System Group Job Support (Suspend/Resume) | 11-1 |
| Examples | 11-1 |
| AS/400 Query | 11-5 |
| AS/400 Business Graphics Utility and Graphical Data Display Manager | |
| Command Interfaces | 11-5 |
| Business Graphics Utility (BGU) | 11-5 |
| Graphical Data Display Manager (GDDM) | 11-6 |
| PC Support/400 | 11-7 |
| Normal Work Station, Database, Communications, and High-Level Language Support | 11-7 |
| Database Files | 11-7 |
| Communications | 11-8 |
| High-Level Language Support | 11-8 |
| OfficeVision/400 V2R2 Application Enabler Support | 11-8 |
| Using Application Enabler Support | 11-9 |
| Setting up Other Office Applications | 11-10 |
| OfficeVision/400 V2R3 Application Enabler Support | 11-12 |
| Using Application Enabler Support | 11-13 |
| Setting up Other Office Applications | 11-14 |
| | |
| Appendix A. Character Set and Code Page Considerations | A-1 |
| Common Character Sets and Code Pages | A-1 |
| CMDCHRID and CHRID Parameter | A-1 |
| Translating Data | A-2 |
| Word Processing Character Set and Code Page Considerations | A-3 |
| Print Options Display | A-3 |
| Override Print File OVRPRTF Command | A-4 |
| Print Files | A-4 |
| QCHRID System Value | A-4 |
| Code Pages Supported by Word Processing | A-5 |
| Work Station Support | A-5 |
| Document Interchange | A-6 |
| | |
| Appendix B. How to Create Control Language Programs and Commands | B-1 |
| Creating Control Language Programs | B-1 |
| Creating a Source Program | B-1 |
| Editing | B-1 |
| Some CL Program Statements | B-1 |
| File Considerations | B-2 |
| Compiling the Program | B-2 |
| Running Your Program | B-3 |
| Debugging Your Program | B-3 |
| Creating Physical Files | B-3 |

| | |
|---|-------|
| Example CL Program | B-4 |
| Creating Control Language Commands | B-6 |
| Creating Source | B-6 |
| Example Command Source | B-7 |
| Creating the Command | B-7 |
| | |
| Appendix C. Office Services Output File Considerations | C-1 |
| Add Calendar Item (ADDCALITM) Output File Considerations | C-1 |
| Display Calendar Authority (DSPCALAUT) Output File Considerations | C-2 |
| Display Calendar Description (DSPCALD) Output File Considerations | C-4 |
| Display Calendar Item (DSPCALITM) Output File Considerations | C-7 |
| Display Directory (DSPDIR) Output File Considerations | C-11 |
| Display Document Library Object Auditing Level (DSPDLOAUD) Output File Considerations | C-39 |
| Display Distribution List (DSPDSTL) Output File Considerations | C-40 |
| Display Document Conversion (DSPDOCCVN) Output File Considerations | C-42 |
| Display Folder (DSPFLR) Output File Considerations | C-43 |
| Folder Object Information | C-43 |
| Document Object Information | C-45 |
| File Document (FILDOC) Output File Considerations | C-47 |
| Fill Form Document (FILLFORM) Output File Considerations | C-47 |
| Print Document (PRTDOC) Output File Considerations | C-49 |
| File Format for PRTFILE Files | C-50 |
| Text Lines | C-58 |
| Query Calendar Item (QRYCALITM) Output File Considerations | C-64 |
| Query Document Library (QRYDOCLIB) Output File Considerations | C-69 |
| Query Distribution (QRYDST) Output File Considerations | C-76 |
| Incoming Distribution Information | C-76 |
| Outgoing Distribution Information | C-82 |
| Receive Distribution (RCVDST) Output File Considerations | C-87 |
| Replace Document (RPLDOC) File Considerations | C-93 |
| Retrieve Document (RTVDOC) Output File Considerations | C-94 |
| Save Document Library Object (SAVDLO) Output File Considerations | C-100 |
| Folder Object Information | C-100 |
| Send Distribution (SNDDST) Output File Considerations | C-104 |
| | |
| Appendix D. Control Language (CL) Commands | D-1 |
| ADDCALITM (Add Calendar Item) Command | D-4 |
| ADDDOCCVN (Add Document Conversion) Command | D-12 |
| ADDOFCAPPE (Add Office Application Entry) Command | D-14 |
| ADDOFCENR (Add Office Enrollment) Command | D-15 |
| ADDTXTIDX (Add Text Index Entry) Command | D-20 |
| CHGCALAUT (Change Calendar Authority) Command | D-23 |
| CHGDOCCVN (Change Document Conversion) Command | D-28 |
| CHGDOCSET (Change Document Set) Command | D-30 |
| CHGDOCTYP (Change Document Type) Command | D-31 |
| CHGOFCA (Change Office Attributes) Command | D-33 |
| CHGOFCAPPD (Change Office Application Description) Command | D-34 |
| CHGOFCAPPE (Change Office Application Entry) Command | D-35 |
| CHGOFENR (Change Office Enrollment) Command | D-36 |
| CHKDOC (Check Document) Command | D-42 |
| CRTCAL (Create Calendar) Command | D-44 |
| CRTDOCSET (Create Document Set) Command | D-50 |
| CRTDOCTYP (Create Document Type) Command | D-51 |

| | |
|--|-------|
| CRTOFCAPPD (Create Office Application Description) Command | D-54 |
| CVTDOC (Convert Document) Command | D-55 |
| DLTCAL (Delete Calendar) Command | D-56 |
| DLTDOCSET (Delete Document Set) Command | D-58 |
| DLTDOCTYP (Delete Document Type) Command | D-59 |
| DLTOFCAPPD (Delete Office Application Description) Command | D-60 |
| DLTOUTMAIL (Delete Outgoing Mail) Command | D-61 |
| DSPCALAUT (Display Calendar Authority) Command | D-62 |
| DSPCALD (Display Calendar Description) Command | D-65 |
| DSPCALITM (Display Calendar Item) Command | D-68 |
| DSPDOCCVN (Display Document Conversion) Command | D-71 |
| DSPIDXSTS (Display Text Index Status) Command | D-73 |
| ENDCALSRV (End Calendar Service) Command | D-74 |
| ENDIDXMON (End Index Monitor) Command | D-75 |
| FILLFORM (Fill Form Document) Command | D-76 |
| MRGDOC (Merge Document) Command | D-79 |
| PAGDOC (Paginate Document) Command | D-84 |
| PRTCAL (Print Calendar) Command | D-85 |
| QRYCALITM (Query Calendar Item) Command | D-94 |
| RMVCALITM (Remove Calendar Item) Command | D-104 |
| RMVDOCCVN (Remove Document Conversion) Command | D-107 |
| RMVOFCAPPE (Remove Office Application Entry) Command | D-108 |
| RMVOFCENR (Remove Office Enrollment) Command | D-109 |
| RMVTEXTIDX (Remove Text Index Entry) Command | D-110 |
| RSTCAL (Restore Calendar) Command | D-113 |
| RTVOFCA (Retrieve Office Attributes) Command | D-117 |
| RTVOFCENR (Retrieve Office Enrollment) Command | D-119 |
| SAVCAL (Save Calendar) Command | D-122 |
| SNDDOC (Send Document) Command | D-127 |
| STRCALSRV (Start Calendar Service) Command | D-128 |
| STRIDXMON (Start Index Monitor) Command | D-129 |
| STROFC (Start Office) Command | D-130 |
| STRPRSDIR (Start Personal Directory) Command | D-131 |
| STRRGZIDX (Start Reorganization of Index) Command | D-132 |
| STRUPDIDX (Start Update of Index) Command | D-133 |
| STRWP (Start Word Processing) Command | D-134 |
| WRKDOCCVN (Work with Document Conversions) Command | D-135 |
| WRKDOCSET (Work with Document Sets) Command | D-136 |
| WRKDOCTYP (Work with Document Types) Command | D-137 |
| WRKOFCAPPD (Work with Office Application Descriptions) Command | D-138 |
| WRKOFCAPPE (Work with Office Application Entries) Command | D-139 |
| WRKTEXTIDX (Work with Text Index) Command | D-140 |
| WRKTEXTPRF (Work with Text Profiles) Command | D-141 |

| | |
|---|------------|
| Appendix E. Hierarchical File System (HFS) and DLS | E-1 |
| Terms and Concepts | E-1 |
| File and Directory Names | E-2 |
| File Types | E-2 |
| Manipulating Files and Directories | E-2 |
| User Enrollment | E-2 |
| Object Authorities | E-2 |
| Directory Entry Attributes | E-3 |
| Standard Attributes | E-3 |
| Interchange Document Profile (IDP) Attributes | E-3 |

| | | |
|--|---|-----|
| | Attributes to Use When Creating Directories and Files | E-5 |
| | Retrieving Attributes | E-7 |
| | HFS Control File System API (QHCTLFS) | E-7 |
| | Bibliography | F-1 |
| | Index | 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 IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

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 Commercial Relations, IBM Corporation, Purchase, NY 10577.

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.

Refer to the "Summary of Changes" on page xvii for a summary of changes made to OfficeVision/400 and how they are described in this publication.

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

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

Programming Interface Information

This publication is intended to help you integrate OfficeVision/400 with business applications in an office environment.

This publication also documents General-Use Programming Interface and Associated Guidance information.

General-Use programming interfaces allow the customer to write programs that obtain the services of OfficeVision/400.

Trademarks and Service Marks

| The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation
| in the United States, in other countries, or in both:

| | |
|------------------------|----------------------------------|
| Application System/400 | OfficeVision/400 |
| APPN | Operating System/2 |
| AS/400 | Operating System/400 |
| C/400 | OS/2 |
| COBOL/400 | OS/400 |
| DisplayWrite | RPG/400 |
| GDDM | SAA |
| IBM | System/370 |
| OfficeVision | Systems Application Architecture |
| OfficeVision/VM | 400 |

| The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other compa-
| nies:

Lotus Lotus Development Corporation

About This Guide

This guide describes the AS/400 command interfaces used to integrate OfficeVision/400 with business applications in an office environment. This guide also provides information about security services and backup and recovery procedures.

You may need to refer to other IBM manuals for more specific information about a particular topic. The *Publications Guide*, GC41-9678, provides information on all the manuals in the AS/400 library.

For a list of related publications, see the Bibliography.

Who Should Use This Guide

This guide is for application programmers designing office applications. Administrators may find the chapters on security services and backup and recovery helpful in maintaining the OfficeVision/400 environment.

You must be familiar with the AS/400 system and the OS/400 program to use this guide.

The following terms have a specific meaning throughout this guide:

| | |
|-------------------------|---|
| Office services | The OfficeVision/400 OS/400 function that is shipped with the AS/400 operating system. Office services are directory services, document distribution services, document library services, security services, and word processing services. They can be accessed with a CL command. |
| OfficeVision/400 | A licensed program that has basic service programs for word processing, storing and retrieving documents, maintaining personal directories, and administering OfficeVision/400. Three optional services are available: text search services, calendar services, and electronic mail services. |
| OS/400 program | The operating system for the AS/400 system. |

Programming Interface Information

This guide contains General-Use Programming Interface and Associated Guidance Information.

Summary of Changes

Chapter 2, "Calendar Services"

- The Print Calendar (PRTCAL) CL command can print monthly calendar views (described in Appendix D, "Control Language (CL) Commands").
- Substitution variables &CURWKNBR, &STRWKNBR, and &ENDWKNBR were added for the calendar week number support.
- QOOCTLCL API has the new parameter, current week, for current week support.

Chapter 4, "Document Distribution Services"

- The output file for the DSPDIR command contains a field for the fax telephone number.

Chapter 6, "Security Services"

- Document and folder event auditing was added.
- Two new commands, DSPDLOAUD and CHGDLOAUD, allow the user to display and change the auditing level for documents and folders. The DSPDLOAUD command can produce an output file containing the auditing level for the specified document library objects.
- The ADDDLOAUT, CHGDLOAUT, and EDTDLOAUT commands allow the user to work with the authority on the root (*ROOT) folder.

Chapter 8, "Word Processing Services"

- Form document support using the FILLFORM command. This allows the user to create a shell document by inserting form field instructions within the document. The FILLFORM command is described in Appendix D, "Control Language (CL) Commands."
- An example application using the form field function is included in this chapter.
- The DSPDOC command includes a new parameter, ALWPRT. This parameter allows the user to specify whether or not to print a document while viewing it.

Chapter 9, "Data Backup, Recovery, and Storage Management"

- When using commitment control and journal management, the user's application now can use the system journal or a separate database file.

Chapter 11, “Enhancing OfficeVision/400”

- The OfficeVision/400 application enabler was enhanced and can use either V2R2 support or V2R3 support. The following commands can be useful in V2R3 support:
 - ADDOFCAPPE Add Office Application Entry
 - CHGDOCSET Change Document Set
 - CHGDOCTYP Change Document Type
 - CHGOFCAPPD Change Office Application Description
 - CHGOFCAPPE Change Office Application Entry
 - CRTDOCSET Create Document Set
 - CRTDOCTYP Create Document Type
 - CRTOFCAPPD Create Office Application Description
 - CVTDOC Convert Document
 - DLTDOCSET Delete Document Set
 - DLTDOCTYP Delete Document Type
 - DLTOFCAPPD Delete Office Application Description
 - RMVOFCAPPE Remove Office Application Entry
 - WRKDOCSET Work with Document Sets
 - WRKDOCTYP Work with Document Types
 - WRKOFCAPPD Work with Office Application Descriptions
 - WRKOFCAPPE Work with Office Application Entries

Detailed information on these commands is found in Appendix D, “Control Language (CL) Commands.”

Appendix E, “Hierarchical File System (HFS) and DLS”

- There is a new appendix which describes using the Hierarchical File System (HFS) with document library services. This appendix describes the parts of the document library services that are relevant to applications.

Chapter 1. Office Command Interfaces

This guide describes the command interfaces of the office services, how to integrate command interfaces into the office product, backup and recovery strategy for office services (including save/restore and journaling), and the function of each of the office services.

The command interfaces for the office services are control language (CL) interfaces. CL can be used in CL programs, in input streams, or interactively on the AS/400* system.

The IBM* SAA* OfficeVision/400* licensed program consists of base office services and optional office services. Base office services consist of the following:

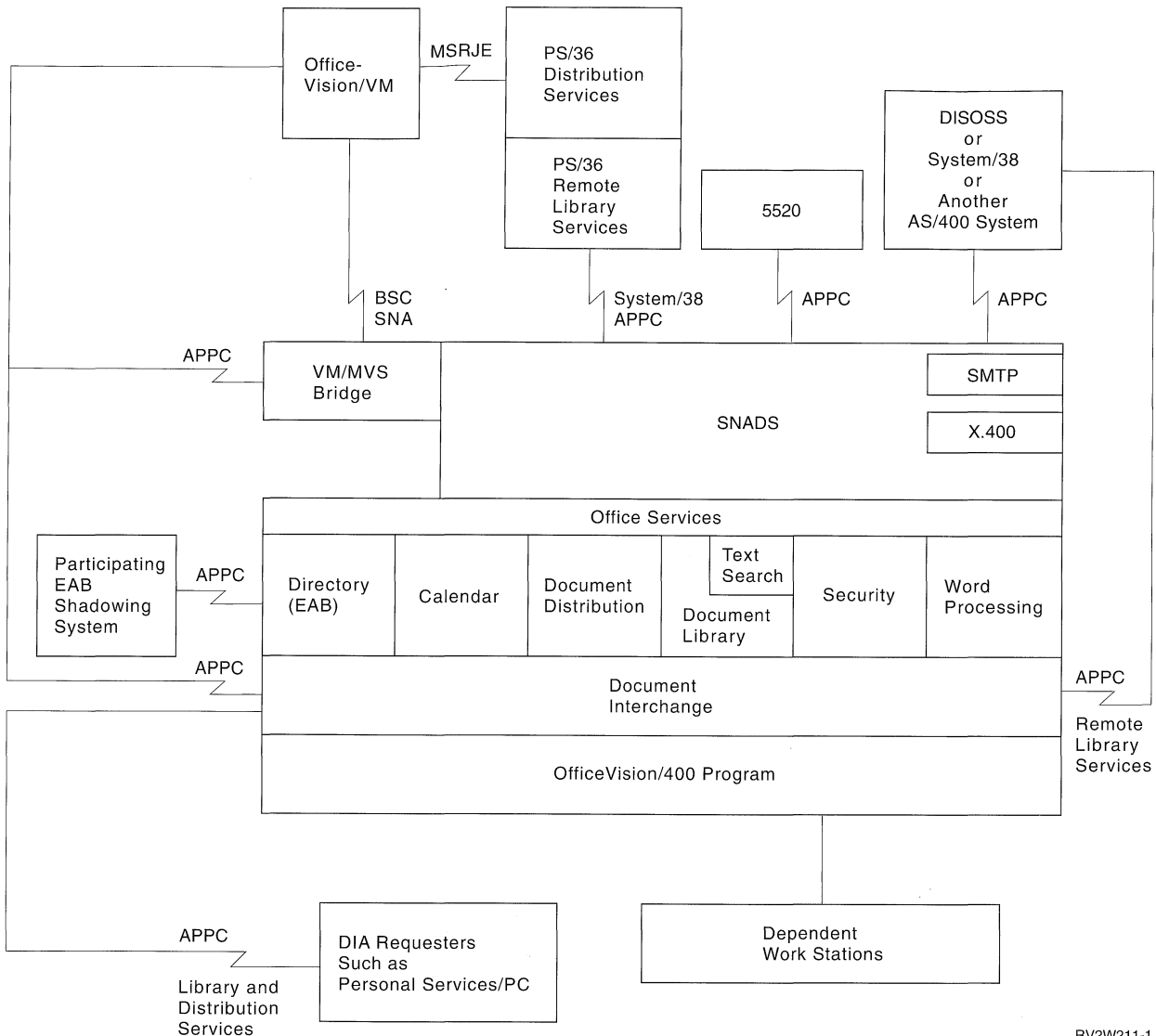
- Directory services, which maintain the system distribution directory. The directory contains information to perform electronic mail distribution, if installed. Refer to Chapter 3, "Directory Services," for more detailed information.
- Document distribution services, which allow users to send and receive documents from other office users. Refer to Chapter 4, "Document Distribution Services," for more detailed information.
- Document library services, which allow users to manage documents, folders, and their related objects. Refer to Chapter 5, "Document Library Services," for more detailed information.
- Security services, which provide security support for office objects, including work on behalf of another user and document and folder security. Refer to Chapter 6, "Security Services," for more detailed information.
- Word processing services, which provide command interface support to some of the word processing functions without having to use OfficeVision/400 menus. Refer to Chapter 8, "Word Processing Services," for more detailed information.

Optional office services consist of the following:

- Calendar services, which allows users to manage calendars and items on calendars. Refer to Chapter 2, "Calendar Services," for more detailed information.
- Electronic mail, which allows users to send electronic mail to another user or group of users. Refer to Chapter 4, "Document Distribution Services," for more detailed information.
- Text search services, which is a document search tool used in word processing services that locates documents by indexed phrases or words in the document. Refer to Chapter 7, "Text Search Services," for more detailed information.

Relationships between Services and Products

Figure 1-1 shows relationships between the various office services and other IBM products.



RV2W211-1

Figure 1-1. Relationships between Office Services and Other IBM Products

OfficeVision/400 accesses local AS/400 document library, document distribution, security, calendar, and word processing services, as well as document library services on a remote System/38, Distributed Office Support System (DISOSS), or other AS/400 systems.

The Personal Services/36, DISOSS, System/38, and 5520 Administrative System have a peer relationship with the AS/400 systems. AS/400 systems can send distributions to and receive distributions from systems that communicate using **SNA distribution services (SNADS)** and **advanced program-to-program communications (APPC)**. SNADS is an asynchronous distribution service that defines a set of rules for receiving, routing, and sending electronic mail in a network of systems. APPC is data communications support that allow programs on an AS/400 system to communicate with programs on other systems that have compatible communica-

tions support. Personal Services/36 can access AS/400 document library services if the user is enrolled on the AS/400 system as an independent (local or remote) work station user.

OfficeVision/VM* can be attached through DISOSS or directly to the AS/400 system. When connected to the AS/400 system, users can exchange documents by using the Virtual Machine/Multiple Virtual Storage (VM/MVS) bridge support.

The OfficeVision/400 support on the AS/400 system is provided by a set of base and optional office services that are part of the Operating System/400* (OS/400*) program, and by other AS/400 licensed programs that provide the user interfaces for various tasks an OfficeVision/400 user needs. In addition to OfficeVision/400, these services are used by other IBM licensed programs, such as PC Support/400 running on systems and work stations attached to an AS/400 system using communications links.

These same services, used by the various IBM products, can be used in applications written by the AS/400 customer and other providers of AS/400 programs.

Interfaces apply to the same set of office services, and can interact with each other and share information. This allows you to integrate your applications with OfficeVision/400 and provide users with access to line of business information through IBM office products, and IBM office information through line of business applications. For example, you can send a distribution from a CL program and use OfficeVision/400 to view, delete, and file it.

Common Considerations for Service Administration

This section describes some of the common considerations for service administration, including enrollment and operational setup. For more information on security considerations, see "Working on Behalf of Another User" on page 6-9 and "Access Control Functions" on page 6-2.

Enrollment

Most control language (CL) commands used to access or administer office services require that the command user be enrolled in the system distribution directory. The **system distribution directory** identifies the name (user ID and address) and system of the authorized users in a SNADS network. **Enrollment** is the process for identifying users to OfficeVision/400. This includes creating a user profile, adding a system distribution directory entry, and calendar information if none of these already exists.

Certain commands do not require that the user be enrolled when the user has *ALLOBJ, *SECADM or *SAVSYS special authority. These commands are listed in "Access Control Functions" on page 6-2. In addition, the Reclaim Document Library Object (RCLDLO) command does not require user enrollment. Refer to the *Distribution Services Network Guide* for more enrollment information.

Operational Setup

The following optional items should be considered before attempting to use office services on an AS/400 system:

- If the system will be in a SNADS network, perform any necessary SNADS configuration. Refer to *Distribution Services Network Guide* for more information.
- Configure APPC devices for the following systems that request remote library services from the AS/400 system:
 - Displaywriter with Electronic Document Distribution (EDD)
 - System/36 with Personal Services/36
 - AS/400 system with OfficeVision/400
 - PS/PC
- Configure APPC devices for the following systems that may be accessed from this system for remote library services:
 - System/38
 - AS/400 system
 - DISOSS

Refer to the *OS/400* Communications Configuration Reference* manual and the *APPC Programmer's Guide* for more information.

- Start QSNADS subsystem before OfficeVision/400 users can perform tasks such as filing or sending documents regardless of whether the AS/400 system is part of a SNADS network. Refer to the Start Subsystem (STRSBS) command in the *CL Reference* manual for more information.

After users are enrolled, and before documents are filed, save the system distribution directory. Save the QUSRSYS library by using the Save Library (SAVLIB) command. This creates a base backup for future recovery use. For more information on backup and recovery, see Chapter 9, "Data Backup, Recovery, and Storage Management."

Office Services Command Interfaces

This section provides an overview of the following office services command interfaces:

- Calendar services
- Directory services
- Document distribution services
- Document library services
- Security services
- Text search services
- Word processing services

The figures on the following pages provide a quick reference to all office command interfaces. See Appendix D, "Control Language (CL) Commands" and other AS/400 programming documents, such as the *CL Reference* manual, and *Database Guide*, for complete descriptions of all CL parameters and file format descriptions. The error messages created by the CL commands can be found in an appendix of the *CL Reference* manual and the *Programming Reference Summary*.

If you use journaling or commitment control in your application, read “Use of Commitment Control and Journal Management” on page 9-10 to understand how office services uses them and how this may affect your application.

The headings in Figure 1-3 on page 1-6 through Figure 1-10 on page 1-14 are defined as follows:

Command The office services CL command being described. See Appendix D, “Control Language (CL) Commands” and the *CL Reference* manual for a detailed description of the command.

Name The name of the CL command.

Database File Formats Used

Some commands create database files as output from the command. These files are described and are expected to be used by customer applications. Other commands accept database files as input.

For those CL commands that either create database files as output or accept database files as input, this column shows the library, file, and format name of the file shipped with the AS/400 program. Programs or users using these CL commands will be working with these same formats through a different user library. Refer to Appendix C, “Office Services Output File Considerations,” for the details of the file format descriptions.

Licensed Program Required

Most of the CL commands need only the OS/400 program on the system to be used. However, a few of the commands require other licensed programs to be installed in order to be used. Definitions of notations in this column are:

| | |
|-------------------|--|
| OfficeVision/400 | OfficeVision/400 must be installed. |
| Calendar Services | OfficeVision/400, with the calendar services option, must be installed. |
| Electronic Mail | OfficeVision/400, with the electronic mail services option, must be installed. |
| Text Search | OfficeVision/400, with the text search option, must be installed. |
| PCS | PC Support/400 must be installed. |
| PCS Org | PC Support/400 organizer must be active. |
| COM I | VM/MVS bridge. |
| blank | Only the OS/400 program is needed to use the command. |

Interactive Only

Most of the CL commands function in either an interactive or a batch environment. The few commands that are operational only in an interactive environment are identified by a Yes in this column.

Calendar Services

The calendar services commands are listed in the following figure:

Figure 1-2. Calendar Services Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|-----------|------------------------------|----------------------------|---------------------------|------------------|
| ADDCALITM | Add Calendar Item | QOFC/QAOYADDITM ADDITM | Calendar Services | |
| CHGCALAUT | Change Calendar Authority | | Calendar Services | |
| CRTCAL | Create Calendar | | Calendar Services | |
| DLTCAL | Delete Calendar | | Calendar Services | |
| DSPCALAUT | Display Calendar Authority | QOFC/QAOYCALAUT CALAUT | Calendar Services | |
| DSPCALD | Display Calendar Description | QOFC/QAOYDSPCAL DSPCAL | Calendar Services | |
| DSPCALITM | Display Calendar Item | QOFC/QAOYDSPITM DSPITM | Calendar Services | |
| ENDCALSRV | End Calendar Service | | Calendar Services | |
| PRTCAL | Print Calendar | | Calendar Services | |
| QRYCALITM | Query Calendar Item | QOFC/QAOYQRYITM QRYCAL | Calendar Services | |
| RMVCALITM | Remove Calendar Item | | Calendar Services | |
| RSTCAL | Restore Calendar | | Calendar Services | |
| SAVCAL | Save Calendar | | Calendar Services | |
| STRCALSRV | Start Calendar Service | | Calendar Services | |

Directory Services

The directory services commands are listed in the following figure:

Figure 1-3 (Page 1 of 2). Directory and Directory Entry Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|-----------|-----------------------------|----------------------------|---------------------------|------------------|
| ADDDIRE | Add Directory Entry | | | |
| ADDDIRSHD | Add Directory Shadow System | | | |
| CHGDIRA | Change Directory Attributes | | | |
| CHGDIRE | Change Directory Entry | | | |

Figure 1-3 (Page 2 of 2). Directory and Directory Entry Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|-----------|------------------------------------|--|---------------------------|------------------|
| CHGDIRSHD | Change Directory Shadow System | | | |
| CPYFRMDIR | Copy from Directory | | | |
| CPYTODIR | Copy to Directory | | | |
| DSPDIR | Display Directory | QSYS/QAOSDIRO .OSDIRE QSYS/QAOSDIRB .OSDIRB QSYS/QAOSDIRF .OSDIRF QSYS/QAOSDIRX .OSDIRX | | |
| ENDDIRSHD | End Directory Shadowing | | | |
| RMVDIRE | Remove Directory Entry | | | |
| RMVDIRSHD | Remove Directory Shadowing | | | |
| RNMDIRE | Rename Directory Entry | | | |
| STRDIRSHD | Start Directory Shadowing | | | |
| WRKDIR | Work with Directory | | | Yes |
| WRKDIRLOC | Work with Directory Locations | | | Yes |
| WRKDIRSHD | Work with Directory Shadow Systems | | | Yes |

The distribution list commands are listed in the following figure:

Figure 1-4 (Page 1 of 2). Distribution List and Distribution List Entry Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|----------|--------------------------------|----------------------------|---------------------------|------------------|
| ADDSTLE | Add Distribution List Entry | | | |
| CRTDSTL | Create Distribution List | | | |
| DLTDSTL | Delete Distribution List | | | |
| DSPDSTL | Display Distribution List | QSYS/QAOSDSTO .OSDSTL | | |
| RMVDSTLE | Remove Distribution List Entry | | | |

Figure 1-4 (Page 2 of 2). Distribution List and Distribution List Entry Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|----------------|------------------------------|-----------------------------------|----------------------------------|-------------------------|
| WRKDSTL | Work with Distribution Lists | | | Yes |

Document Distribution Services

The document distribution commands are listed in the following figure:

Figure 1-5. Document Distribution Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|---------|---------------------------------|--|---------------------------|------------------|
| CHGDSTD | Change Distribution Description | | | |
| DLTDST | Delete Distribution | | | |
| QRYDST | Query Distribution | QSYS/QAOSILIN .OSLIN QSYS/QAOSILOT .OSLOUT | | |
| RCVDST | Receive Distribution | QSYS/QAOSIRCV .OSRCVD | | |
| SNDDOC | Send Document | | OfficeVision/400 | Yes |
| SNDDST | Send Distribution | QSYS/QAOSIRCV .OSRCVD QSYS/QAOSIRTV .OSRTVD | | |

Document Library Services

The document library services commands are listed in the following figure:

Figure 1-6 (Page 1 of 2). Document Library Services Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|-----------|--------------------------------------|--|---------------------------|------------------|
| CHGDOCD | Change Document Description | | | |
| CHKDLO | Check Document Library Object | | | |
| CPYDOC | Copy Document | | | |
| CPYFRMPCD | Copy From PC Document | See note ¹ | PCS | |
| CPYTOPCD | Copy To PC Document | See note ¹ | PCS | |
| CRTFLR | Create Folder | | | |
| DLTDLO | Delete Document Library Object | | | |
| DMPDLO | Dump Document Library Object | | | |
| DSPFLR | Display Folder | QSYS/QADSPFLR .FLRDTL QSYS/QADSPDOC .DOCCTL | | |
| DSPDLONAM | Display Document Library Object Name | | | |

Figure 1-6 (Page 2 of 2). Document Library Services Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|---------------------|---------------------------------------|--|---------------------------|------------------|
| FILDOC | File Document | QSYS/QAOSIRCV .OSRCVD QSYS/QAOSIRTV .OSRTVD | | |
| MOVDOC | Move Document | QSYS/QAOSIRCV .OSRCVD QSYS/QAOSIRTV .OSRTVD | | |
| QRYDOCLIB | Query Document Library | QSYS/QAOSIQDL .OSQDL | | |
| RCLDLO ² | Reclaim Document Library Object | | | |
| RGZDLO | Reorganize Document Library Object | | | |
| RNMDLO | Rename Document Library Object | | | |
| RPLDOC | Replace Document | QSYS/QAOSIRCV .OSRCVD QSYS/QAOSIRTV .OSRTVD | | |
| RSTDLO ³ | Restore Document Library Object | | | |
| RTVDLONAM | Retrieve Document Library Object Name | | | |
| RTVDOC | Retrieve Document | QSYS/QAOSIRTV .OSRTVD | | |
| SAVDLO ³ | Save Document Library Object | QSYS/QAOJSAVO .OJSDLO | | |

Notes:

1. User-defined source or data files are used. See the *PC Support/400 DOS Installation and Administration Guide* or the *PC Support/400 OS/2 Installation and Administration Guide* for information on the format of PC data used by these commands.
2. See Chapter 9, "Data Backup, Recovery, and Storage Management," for information on the RCLDLO command.
3. For the RSTDLO and SAVDLO commands, see Chapter 9, "Data Backup, Recovery, and Storage Management," the *CL Reference* manual, and the *Basic Backup and Recovery Guide*.

Security Services

The security services commands are listed in the following figure:

Figure 1-7. Security Services Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|------------|--------------------------------|----------------------------|---------------------------|------------------|
| ADDDLOAUT | Add DLO Authority | | | |
| CHGDLOAUD | Change DLO Auditing level | | | |
| CHGDLOAUT | Change DLO Authority | | | |
| CHGDLOOWN | Change DLO Owner | | | |
| DSPAUTLDLO | Display Authorization List DLO | | | |
| DSPDLOAUD | Display DLO Auditing level | | | |
| DSPDLOAUT | Display DLO Authority | | | |
| DSPUSRPMN | Display User Permission | | | |
| EDTDLOAUT | Edit DLO Authority | | | Yes |
| GRTUSRPMN | Grant User Permission | | | |
| RMVDLOAUT | Remove DLO Authority | | | |
| RVKUSRPMN | Revoke User Permission | | | |

Text Search Services

The text search services commands are listed in the following figure:

Figure 1-8 (Page 1 of 2). Text Search Services Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|------------|---------------------------|----------------------------|---------------------------|------------------|
| ADDTXTIDXE | Add Text Index Entry | | Text Search | |
| DSPIDXSTS | Display Text Index Status | | Text Search | Yes |
| ENDIDXMON | End Index Monitor | | Text Search | |
| RMVTXTIDXE | Remove Text Index Entry | | Text Search | |
| STRIDXMON | Start Index Monitor | | Text Search | |

Figure 1-8 (Page 2 of 2). Text Search Services Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|----------------|------------------------------------|-----------------------------------|----------------------------------|-------------------------|
| STRRGZIDX | Start Reorganiza- tion of Index | | Text Search | |
| STRUPDIDX | Start Update of Index | | Text Search | |
| WRKTXIDX | Work with Text Index | | Text Search | Yes |

Word Processing Services

The word processing services commands are listed in the following figure:

Figure 1-9. Word Processing Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|------------|--------------------------------|----------------------------|---------------------------|------------------|
| CHKDOC | Check Document | | OfficeVision/400 | |
| CRTDOC | Create Document | | OfficeVision/400 or PCS | Yes |
| CRTSPADCT | Create Spelling Aid Dictionary | See note ¹ | | |
| DLTSPADCT | Delete Spelling Aid Dictionary | | | |
| DSPDOC | Display Document | | OfficeVision/400 or PCS | Yes |
| DSPHLPDOC | Display Help Document | | | Yes |
| EDTDOC | Edit Document | | OfficeVision/400 or PCS | Yes |
| MRGDOC | Merge Document | | OfficeVision/400 | |
| PAGDOC | Paginate Document | | OfficeVision/400 | |
| PRTDOC | Print Document | QSYS/QAOPOUFL .RECORD | | |
| PRTSWL | Print Stop Word List | | | |
| RTVSWLSRC | Retrieve Stop Word List Source | | | |
| STRWP | Start Word Processing | | OfficeVision/400 | Yes |
| WRKDOC | Work with Documents | | See note ² | Yes |
| WRKDOCprtq | Work with Document Print Queue | | | Yes |
| WRKFLR | Work with Folders | | See note ³ | Yes |
| WRKTXTPRF | Work with Text Profiles | | OfficeVision/400 | Yes |

Notes:

1. Requires an input source file created using the Start SEU (STRSEU) command. The source file contains the words for the dictionary being created. You may have one word per source file line, or multiple words separated by at least one blank per source file line. You may include one or more hyphenation marks in words to indicate desired hyphenation points.
2. There is no licensed program required to use the WRKDOC command. However, there are some options on subsequent displays presented by this command, such as view and revise a document, that require PC Support/400 or OfficeVision/400 (or both) for processing.
3. There is no licensed program required to use the WRKFLR command. However, there are some options on subsequent displays presented by this command, such as view and revise a document, that require PC Support/400 or OfficeVision/400 for processing.

Miscellaneous Services

The miscellaneous services commands are listed in the following figure:

Figure 1-10 (Page 1 of 2). Miscellaneous Services Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|------------|---------------------------------------|----------------------------|---------------------------|------------------|
| ADDDOCCVN | Add Document Conversion | | OfficeVision/400 | |
| ADDOFCAPPE | Add Office Application Entry | | OfficeVision/400 | |
| ADDOFCENR | Add Office Enrollment | | OfficeVision/400 | |
| CFGRPDS | Configure VM/MVS Bridge | | COM I | Yes |
| CHGDOCCVN | Change Document Conversion | | OfficeVision/400 | |
| CHGDOCSET | Change Document Set | | OfficeVision/400 | |
| CHGDOCTYP | Change Document Type | | OfficeVision/400 | |
| CHGOFCA | Change Office Attributes | | OfficeVision/400 | |
| CHGOFCAPPD | Change Office Application Description | | OfficeVision/400 | |
| CHGOFCAPPE | Change Office Application Entry | | OfficeVision/400 | |
| CHGOFCENR | Change Office Enrollment | | OfficeVision/400 | |
| CRTDOCSET | Create Document Set | | OfficeVision/400 | |
| CRTDOCTYP | Create Document Type | | OfficeVision/400 | |
| CRTOFCAPPD | Create Office Application Description | | OfficeVision/400 | |
| CVTDOC | Convert Document | | OfficeVision/400 | |
| DLTDOCSET | Delete Document Set | | OfficeVision/400 | |
| DLTDOCTYP | Delete Document Type | | OfficeVision/400 | |
| DLTOFCAPPD | Delete Office Application Description | | OfficeVision/400 | |
| DLTOUTMAIL | Delete Outgoing Mail | | Electronic Mail | |

Figure 1-10 (Page 2 of 2). Miscellaneous Services Commands

| Command | Name | Database File Formats Used | Licensed Program Required | Interactive Only |
|------------|---|----------------------------|---------------------------|------------------|
| DSPDOCCVN | Display Document Conversion | | OfficeVision/400 | |
| RMVDOCCVN | Remove Document Conversion | | OfficeVision/400 | |
| RMVOFCAPPE | Remove Office Application Entry | | OfficeVision/400 | |
| | | | | |
| RMVOFCENR | Remove Office Enrollment | | OfficeVision/400 | |
| RTVOFCA | Retrieve Office Attributes | | OfficeVision/400 | |
| RTVOFCENR | Retrieve Office Enrollment | | OfficeVision/400 | |
| STROFC | Start OfficeVision/400 | | OfficeVision/400 | Yes |
| WRKDOCCVN | Work with Document Conversions | | OfficeVision/400 | Yes |
| WRKDOCLIB | Work with Document Library | | | Yes |
| | | | | |
| WRKDOCSET | Work with Document Sets | | OfficeVision/400 | |
| | | | | |
| WRKDOCTYP | Work with Document Types | | OfficeVision/400 | |
| | | | | |
| WRKOFCAPPD | Work with Office Application Descriptions | | OfficeVision/400 | |
| | | | | |
| WRKOFCAPPE | Work with Office Application Entries | | OfficeVision/400 | |
| | | | | |

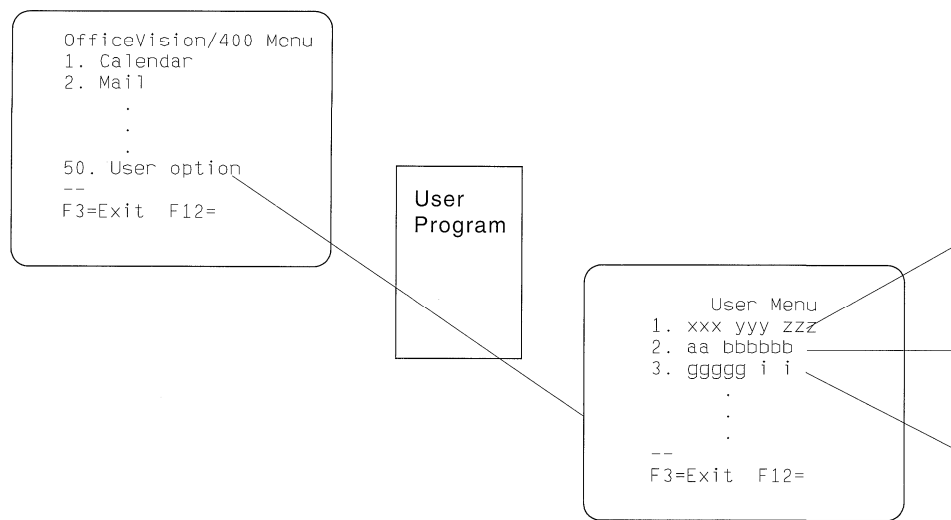
Integrating Applications with OfficeVision/400 Using Commands

This section discusses methods of integrating applications with OfficeVision/400 and considerations for each method. To set up the following examples, see “Start Office (STROFC) Command” on page 10-2 and “Example Applications” on page 10-10.

Customizing the Main Menu

Use option 50 to define a path to a user-written program. It is accessible directly from OfficeVision/400. This user-written program can be very flexible and perform any number of tasks, including presenting another menu. The OfficeVision/400 administrator may set up an option 50 for a user by selecting the *Enrollment information* option presented by the Administration function of the OfficeVision/400 main menu.

Customizing the menu is shown in the following figure:



RSLW126-2

label='Customizing the main menu'.

Figure 1-11. Customizing the Main Menu

Considerations for customizing the menu this way include:

- Direct access from OfficeVision/400.
- Suspend and Resume capability using the ATTN key.
- Ease of use with an enrollment option.
- New *mail* message on the OfficeVision/400 main menu.
- OfficeVision/400 functions cannot be called from applications, and return to the main menu is required.

Using OfficeVision/400 Functions with User Application Menus

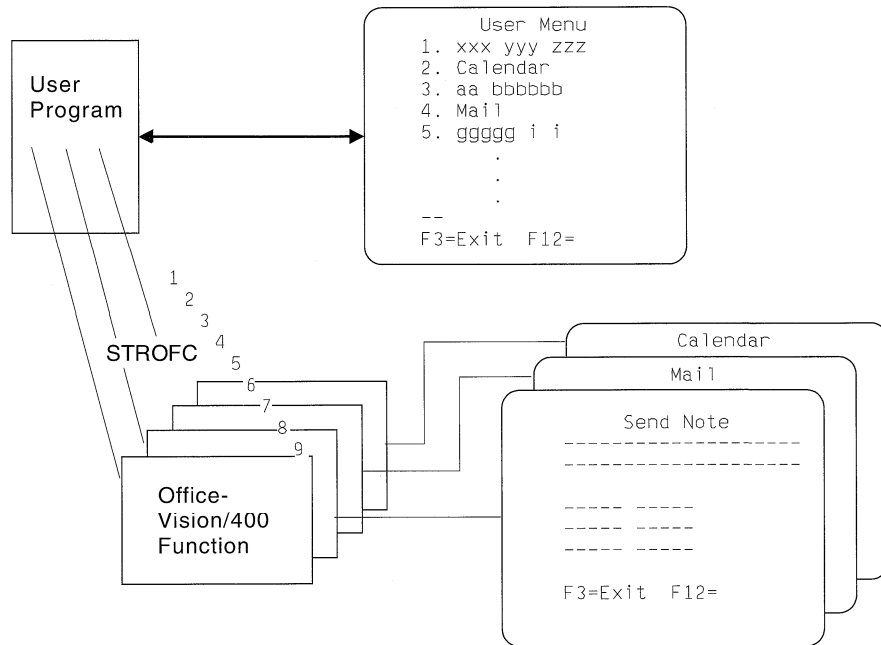
When a user is enrolled in OfficeVision/400, all of its functions are available for use, depending on the user's authority.

If the preference is to limit some or all users to just a few of the OfficeVision/400 functions, the customer can write an application for producing a menu. When a

user signs on the system, the menu that appears is the one produced by the customer application.

Certain options on the customer menu could be functions from the OfficeVision/400 licensed program. When a user selects one of those options, the customer application can get to that OfficeVision/400 function by using the Start OfficeVision/400 (STROFC) command with one parameter that corresponds to the OfficeVision/400 main menu option number.

Using OfficeVision/400 with user application menus is shown in the following figure:



RSLW127-1

label='Using OfficeVision/400 with Application Menus'.

Figure 1-12. Using OfficeVision/400 with User Application Menus

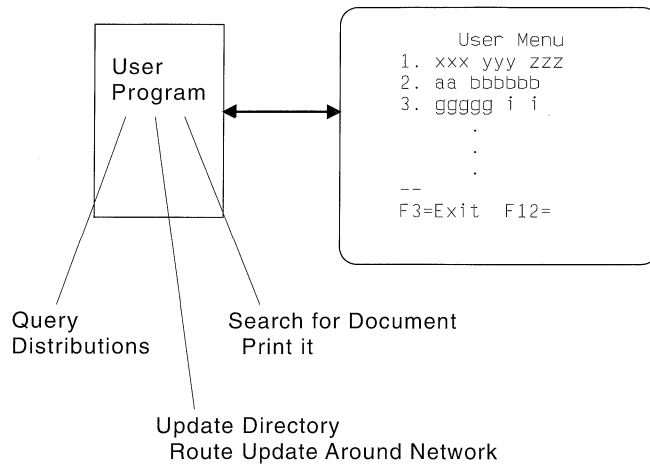
Considerations for this method include:

- Better integration with application (go directly from application to OfficeVision/400 function).
- May be more efficient.
- Provides a method to limit office function for some users.
- User can program a command line on menus.
- No automatic suspend/resume support (code must be written to get suspend/resume). See "Operating System Group Job Support (Suspend/Resume)" on page 11-1 for techniques on this procedure.
- User must program support to send new *mail waiting* message.

Using Office Services with User Applications

You can process some of the office functions through command interfaces. By using the command interfaces, you can closely integrate to specific customer needs. For example, you can set up a user menu to call a user program that accesses the commands. This user menu can be a combination of calls to options in OfficeVision/400 and calls to your own programs. If you do not have OfficeVision/400, this can be a way of setting up a user menu that calls programs to do office processing.

The following figure shows office services with user applications:



RSLW128-0

label='Office Services with User Applications'.

Figure 1-13. Office Services with User Applications

A consideration with this method is that it is not limited to an existing OfficeVision/400 function. This allows for:

- Closer integration to specific customer needs
- Standard CL interfaces, high-level language (HLL) program access, and tool support

Chapter 2. Calendar Services

This chapter discusses calendar services and the control language (CL) commands available to maintain calendar services. It also discusses using calendar user-defined functions and describes the Control Calendar application program interface (API). Calendar services is an optional OfficeVision/400 service that must be installed on your system before you can use the calendar functions.

Calendar Services Overview

Calendar services allows users to maintain electronic calendars and items scheduled on calendars. A calendar is an internal system object that is contained in IBM-supplied database files. An item is any event, meeting, reminder, meeting entry, job, or System/36 procedure that is scheduled on a given calendar. Items are also contained in IBM-supplied database files.

A **calendar** is identified by a 10-character name and the user ID and address of the owner. A calendar name must not start with a blank or an asterisk (*). Each user enrolled in OfficeVision/400 may also have a assigned **scheduling calendar**.

A **calendar group** is identified by a 10-character name and the user ID and address of the owner. A calendar group name must not start with a blank or an asterisk (*). A calendar group contains a collection of calendars used to work with several calendars at once, instead of one calendar at a time. A calendar group may contain both local and remote calendars. Remote calendars may or may not exist on the remote system and are not checked by calendar services.

A **calendar item** is an event, meeting, meeting entry, reminder, job, or System/36 procedure which is scheduled on a calendar at a given date, and possibly, time. Each item is unique and is identified by an item identifier. An item identifier along with a calendar uniquely identifies all items on the system. When used in CL commands, an item identifier allows a single calendar item to be processed.

A set of authorities can be specified for each calendar. The **calendar authorities** list user IDs and addresses and specific authorities each user is granted. A special entry also exists for *PUBLIC, which specifies the authorities any user which is not listed has to the calendar. The authorities granted apply to each of the three calendar item security levels: unclassified, confidential, and personal.

Calendar services allows you to:

- Create, delete, display, save, restore, and print calendars
- Change and display calendar authorities
- Add, remove, query, and display calendar items
- Start and end the calendar service program
- Create calendar user-defined functions to use an application from within calendar services

Calendar Services Objects

This section discusses calendar files, calendar group files, calendar item files, calendar authority files, calendar services output files, and naming conventions.

All calendar services files, except model output files, are defined and shipped with the system in library QUSRSYS. Calendar services files use the following naming conventions:

| | |
|------------------|---|
| QAOCCL5 | Calendar physical file |
| QAOCCL5x | Calendar logical files |
| QAOCGM5 | Calendar group member physical file |
| QAOCGM5x | Calendar group member logical files |
| QAOCGR5 | Calendar group physical file |
| QAOCGR5x | Calendar group logical files |
| QAOCIT5 | Calendar item physical file |
| QAOCIT5x | Calendar item logical files |
| QAOCIN5 | Calendar item invitee physical file |
| QAOCIN5x | Calendar item invitee logical files |
| QAOCXT5 | Calendar item text physical file |
| QAOC AU5 | Calendar authority physical file |
| QAOC AU5x | Calendar authority logical files |
| QAOCMA5 | Calendar meeting authorizations physical file |
| QAOCMA5x | Calendar meeting authorizations logical files |

where *x* is a single alphabetic character. These files should not be directly accessed or deleted. See Chapter 9, "Data Backup, Recovery, and Storage Management" for information on saving and restoring calendar services information.

The naming convention for calendar services model output files in library QOFC is QAOYxxxx.

Note: Customer programs must not access the calendar services files directly, because it is an IBM-reserved right to change the file format or the method or location of storing calendar services data.

Common CL Command Parameter Information

This section discusses the CL parameters used in calendar services.

Library Parameters

The following special values are allowed for the library name on most commands:

- *LIBL/** Search the library list of the current job for the object. The first object found that matches the name and object type is used.
- *CURLIB/** Use the current library in the library list. Use the CHGCURLIB command to change the current library.

Date Parameters

The date parameters are syntax-checked by the system. This means that the date must be in the form specified by the QDATFMT system value. The date separators are optional, but if specified, the separator must be the value in the system value QDATSEP. If the separator characters are used, the date must be enclosed in apostrophes.

Note: If prompting for commands, the system adds the apostrophes. There is not enough room for you to add the apostrophes on the prompt screen.

Time Parameters

The time parameters are syntax-checked by calendar services. All times must be entered in 24-hour format. Allowed time formats are:

- H
- HH
- HMM
- HHMM
- H:MM
- HH:MM

If a time separator is used, the time must be enclosed in apostrophes.

Note: If prompting for commands, the system adds the apostrophes.

ITMID Parameters

The item identifier uniquely identifies an item on a calendar. When used with a calendar name, the system can validate that the item identifier and calendar name both refer to the same calendar. Item identifiers and their corresponding calendars can be found in the output files of ADDCALITM and QRYCALITM.

CAL Parameter

The CAL parameter is available on all the calendar CL commands. This parameter determines which calendars are to be processed by the command. The CAL parameter searches and returns all calendars that match the calendar names and owners specified. Not all the CL commands support the same combinations for the calendar name and owner.

GRPCALS Parameter

The GRPCALS parameter allows you to specify a single local calendar group by name and, optionally, by owner. You must specify CAL(*GRP) when the GRPCALS parameter is used. All calendars in the specified calendar group are processed by the command.

Authority Parameters

Calendars have both public and private authorities. If you are not specifically authorized to a calendar, you are given the public authority.

The authority parameters on the calendar CL commands support the following values:

- *EXCLUDE** You do not have authority to see any information of the calendar items.
- *TIMES** You can see the times of the calendar items, but cannot see any of the text associated with the items.
- *USE** You can see the times and the text of the calendar items, but are not allowed to add or change any of the items.
- *ADDITEM** You can see the times and the text of the calendar items, and add items to the calendar. In addition, you are allowed to change or remove the items you have added.
- *CHANGE** You can see the times and the text of the calendar items, add items to the calendar, and change or remove any item on the calendar. You cannot change or delete the calendar itself or change the list of authorized users.

The authority values apply separately to each of the calendar item security levels: unclassified, confidential, and personal. The authority you have to unclassified items on the calendar can be different from the authority you have to personal items. For example, *CHANGE authority to unclassified items means you can change or remove any unclassified item, but does not mean you can change or remove any personal item.

If you specify *TIMES authority or a greater authority value for any of the security levels, you automatically are given *TIMES authority to all the levels. This allows you to do a valid free time search. You must have at least *TIMES authority to do a free time search.

If you specify *USE authority or a greater authority value for personal or confidential items, your unclassified authority is automatically set to the highest of the personal, confidential, or the unclassified authority specified.

Character Set and Code Page Considerations

All output files from calendar services contain fields that are translatable and correspond with fields in the character set and code page used in the output file. An attempt is made to translate all fields to the system character set and code page, but in cases where the translation fails, the fields are correctly tagged in the output file.

CMDCHRID Parameter

This parameter indicates the character set and code page of other parameters which are translatable. Since the default is *SYSVAL, meaning use the system value QCHRID, it is up to you to specify *DEV D if your device is different than the system device when entering commands interactively.

Other Calendar Services Topics

This section discusses security considerations, including restrictions which apply to users, security officers (*ALLOBJ), security administrators (*SECADM), and users with *SAVSYS special authority.

Security Considerations

All calendar services control language (CL) commands used within a program adopt the authority of that program. Thus, a program owned by a user with *SECADM authority that runs under the authority of the program owner, provides security administrator authority to any calendar services commands used. All calendar services commands use *ALLOBJ and *SECADM authority as equals. Thus, a user with *ALLOBJ special authority or *SECADM special authority are both handled as security administrators when using calendar services commands.

Calendar Services Control Language Commands

See Appendix D, "Control Language (CL) Commands" for detailed information about command parameters for the CL commands discussed in this section. See the *Programming Reference Summary* for information on error messages signaled by the CL commands.

The following commands are described in this section:

- Add Calendar Item (ADDCALITM)
- Change Calendar Authority (CHGCALAUT)
- Create Calendar (CRTCAL)
- Delete Calendar (DLTCAL)
- Display Calendar Authority (DSPCALAUT)
- Display Calendar Description (DSPCALD)
- Display Calendar Item (DSPCALITM)
- End Calendar Service (ENDCALSRV)
- Print Calendar (PRTCAL)
- Query Calendar Item (QRYCALITM)
- Remove Calendar Item (RMVCALITM)
- Restore Calendar (RSTCAL)
- Save Calendar (SAVCAL)
- Start Calendar Service (STRCALSRV)

All of these commands may be used interactively, in batch, or within a program.

Add Calendar Item (ADDCALITM) Command

The Add Calendar Item (ADDCALITM) command adds one or more items to one or more calendars. The calendars must be local calendars unless the item is a meeting. The items added can be any of the following:

- Event** An item related to a single calendar. An event has a starting and ending time. The ADDCALITM command allows an event to be added to multiple calendars.
- Meeting** An item related to multiple calendars. A meeting has a starting and ending time. The meeting item itself is not on any calendar, but meeting entries can be added to each calendar specified (these are the meeting invitees). The meeting entries are all related to the

central meeting item, but are used to hold unique information for each invitee (comments about the meeting or attendance status). The central meeting item contains common information such as date, time, subject, and so on. If you are not authorized to an invitees' calendar or the invitee is remote, then a meeting notice is sent to the invitee.

Reminder An item related to a single calendar. A reminder is associated with a day, but does not have a start or end time. It does not take up any time on the calendar. The ADDCALITM command allows a reminder to be added to multiple calendars.

Job An item related to a single calendar. A job has a start time that specifies when the job should run. The ADDCALITM command allows a job to be added to multiple calendars.

System/36 procedure

An item related to a single calendar. A System/36 procedure has a start time that specifies when the procedure should run. The ADDCALITM command allows a procedure to be added to multiple calendars.

An item can recur on a calendar (for example, every Monday). The system provides methods for working with recurring items as a group. For example, you can remove, copy, and change parts of recurring items as a group.

Security Considerations

A user must have at least *ADDITEM authority to add items to a calendar. When adding a meeting with less than *ADDITEM authority, a meeting entry is not placed on the calendar, but a notice is sent. The user must have *CHANGE authority to add items with a status of *CONFIRMED. Otherwise, the status is reset to *TENTATIVE.

Some Ways to Use This Command

To add an item to your scheduling calendar, type the following command. The event is confirmed from 8 a.m. to 10 a.m. on February 27, 1990. The text for the event is Doctor appointment:

```
ADDCALITM CAL(*SCHED) DATE(022790) TIME(('8:00' '10:00')) ITMTYP(*EVENT)
      RMDMSG(*NO) ITMSTS(*CONFIRMED) TEXT('Doctor appointment')
```

To add a meeting with all calendars listed in the calendar group DEPT54P as meeting invitees, type the following command. The meeting is held weekly on Tuesday (January 2, 1990, is a Tuesday) from 1 p.m. to 2 p.m. The subject of the meeting is Dept mtg and it is held in Bob's office. The purpose of the meeting is to discuss scheduling.

```
ADDCALITM CAL(*GRP) GRPCALS(DEPT54P) DATE(010290) TIME(('13:00' '14:00'))
      RCR(*WEEKLY) ENDDATE(123190) ITMTYP(*MEETING) MTGSUB('Dept mtg')
      MTGPLC('Bobs office') TEXT('Discuss scheduling')
```

To add a reminder to four users' scheduling calendars, reminding them that timecards must be turned in, type the following command. The results are written to output file ADDOUT in library QGPL, member ADDMBR1. The output file records contain the item identifiers and calendar names for each reminder that was added.

```
ADDCALITM CAL((*SCHED MIKE ROCH) (*SCHED JOE ROCH)
(*SCHED DARLA ROCH) (*SCHED LAURIE ROCH)) ITMTYP(*REMINDER)
DATE('3/6/90') TEXT('Turn in timecards') RMDMSG(*YES)
RMDMSGTIME(*ITEMDATE '7:00') OUTFILE(QGPL/ADDOUT) OUTMBR(ADDMBR1)
```

Rules

The following rules apply to the ADDCALITM command:

- The job command (JOBCMD) parameter is defined as a command string and cannot be specified as a variable.
- When specifying an item identifier (ITMID) as input to this command, the item identifier must not refer to an existing item. This is intended to allow an application to reuse an old item identifier or to extend a recurring sequence.
- All jobs and System/36 procedures added for past dates are not run.
- Any items added for past dates with RMDMSG(*CALENDAR) specified, are added with reminder message set to *NO, regardless of what the calendar specifies.
- Users whose enrollment record specifies ALWCALJOB(*NO) cannot add jobs or procedures.
- No conflict checking is done when the item is added. If conflict checking is needed, use the QRYCALITM command before adding the item.

Hints and Techniques

The following hints and techniques can help you use the ADDCALITM command:

- The last five digits of an item identifier represent the occurrence number. Recurring items have the same first 15 digits. To add a new occurrence, specify an item identifier as input and increase the last five digits. The maximum number of occurrences is 500, so you cannot increase the last five digits beyond '00500'.
- After adding a meeting, you must use the interactive OfficeVision/400 calendar product to make any changes or to delete the meeting.
- A change operation can be accomplished by removing an item and adding it again with the same item identifier given as input (this applies to nonmeeting items only).

Change Calendar Authority (CHGCALAUT) Command

The Change Calendar Authority (CHGCALAUT) command allows you to change the public and private authorities to one or more calendars.

Security Considerations

The calendar authorities can be changed for a distribution list, list of users, or a reference calendar. You must be the owner of the calendar or have *SECADM or *ALLOBJ authority to use this command.

Some Ways to Use This Command

To change the public authority for your scheduling calendar to *EXCLUDE, type:

```
CHGCALAUT CAL(*SCHED) USER(*PUBLIC) UNCAUT(*EXCLUDE)
      CONAUT(*EXCLUDE) PRSAUT(*EXCLUDE)
```

To change the unclassified authority for the calendars listed in the calendar group DEPT to *CHANGE for all users listed in the distribution list DEPT DSTL, type:

```
CHGCALAUT CAL(*GRP) GRPCALS(DEPT) DSTL(DEPT DSTL) UNCAUT(*CHANGE)
```

The personal and confidential authorities are changed to *TIMES if the current authority is *EXCLUDE (to allow for a valid free time search); otherwise, they are not changed.

To change all the calendars owned by user PAT to have the same authorities as the scheduling calendar for PAT, type the following command. Any private authorities already granted to calendars that are not in the scheduling calendar are not changed.

```
CHGCALAUT CAL(*ALL PAT ROCH) REFCAL(*SCHED PAT ROCH)
```

To change the confidential and personal authorities to your scheduling calendar to *ADDITEM for local users PAT and MITCH, type the following command. The unclassified authority is changed to *ADDITEM if it is not already at least *ADDITEM.

```
CHGCALAUT CAL(*SCHED) USER((MITCH ROCH) (PAT ROCH))
      CONAUT(*ADDITEM) PRSAUT(*ADDITEM)
```

To delete the private authority record for user MITCH from your scheduling calendar, type the following command. MITCH receives the same authority as *PUBLIC until another private authority for MITCH is granted.

```
CHGCALAUT CAL(*SCHED) USER(MITCH ROCH) UNCAUT(*PUBLIC)
      CONAUT(*PUBLIC) PRSAUT(*PUBLIC)
```

Rules

The following rules apply to the CHGCALAUT command:

- Local and remote users in the system distribution directory can be authorized to a calendar. Users not in the system distribution directory can also be authorized to a calendar, but an informational message is issued warning that the user ID and address might be incorrect.
- The authorities are adjusted as described in “Authority Parameters” on page 2-4.

Create Calendar (CRTCAL) Command

The Create Calendar (CRTCAL) command allows you to create a new calendar.

Security Considerations

When you create a calendar, you can specify the manager of the calendar, the working times for the calendar, the notification mode for entries on the calendar, and description of the calendar. You can change these values later, if necessary. If you have *SECADM or *ALLOBJ authority, you can specify an owner other than yourself.

Some Ways to Use This Command

To create a calendar with all the defaults, type:

```
CRTCAL CAL(MIKE)
```

If MIKE is the current user signed on, this makes MIKE the owner and the manager of the calendar. The public does not have any authority to the items on the calendar. Notification is not sent to the manager or the owner. The working times are set to the default, 8 a.m. to 5 p.m., Monday through Friday. The description of the calendar is the owners' directory entry description.

To create a calendar with yourself as the owner, type:

```
CRTCAL CAL(MIKE) TEXT('My calendar') MGR(SUE ROCH) NFYMGR(*YES)
NFYMOD(*MAIL) PUBUNCAUT(*USE) WEDWRKTIME((*NONE))
```

This assigns Sue to be the manager of the calendar. Notification is sent to the manager through mail and the message queue. The public is given authority to look at the unclassified items on the calendar. The public authority for confidential and personal items is *TIMES. The working times are set to 8 a.m. to 5 p.m. on Monday, Tuesday, Thursday, and Friday (the default) and set to NONE for Saturday, Sunday, and Wednesday. The description of the calendar is "My Calendar".

To create a calendar for a conference room, type:

```
CRTCAL CAL(CONFROOM SUE ROCH) TEXT('Calendar for the room')
PUBUNCAUT(*ADDITEM) RMDLDTIME(30) SNDMSG(*YES)
SATWRKTIME((8:00 17:00)) SUNWRKTIME((8:00 17:00))
```

The calendar is owned by SUE. The public is authorized to add unclassified items to the calendar. This gives the public *ADDITEM authority to unclassified items, and *TIMES authority to confidential and personal items. If you have ADDITEM authority to all security levels (for example, unclassified, confidential, and personal), then you must also specify PUBCONAUT(*ADDITEM) and PUBPRSAUT(*ADDITEM). This gives you *ADDITEM authority to unclassified, confidential, and personal items. The owner (SUE) is automatically sent a reminder message 30 minutes before each item on the calendar occurs. Also, the conference room is available from 8 a.m. to 5 p.m. every day of the week, including Saturday and Sunday.

To create a calendar from another calendar, type:

```
CRTCAL CAL(SUE) FROMCAL(*SCHED *CURRENT)
```

Calendar SUE is created with the same parameters that the scheduling calendar of the current user has. These parameters can be overridden by specifying the parameter on the CRTCAL command directly. For example, PUBUNCAUT(*TIMES) overrides the unclassified authority for the current user's scheduling calendar.

Rules

The following rules apply to the CRTCAL command:

- The owner and manager of the calendar being created must be enrolled in OfficeVision/400.

- The FROMCAL parameter must specify a calendar to which you have at least *USE authority. A user with *SECADM or *ALLOBJ special authority can specify any calendar.
- The notify manager mode (NFYMOD parameter) can only be specified if the NFYMGR parameter is set to *YES.
- The authorities are adjusted as described in the section “Authority Parameters” on page 2-4.
- The working times are adjusted to the nearest 15 minute interval. For example, 8:01 is adjusted to 8:00 and 8:08 is adjusted to 8:15.

Delete Calendar (DLTCAL) Command

The Delete Calendar (DLTCAL) command allows you to delete one or more calendars.

Security Considerations

You must be the owner of the calendar or have *SECADM or *ALLOBJ special authority to delete a calendar. When a calendar is deleted, all items on that calendar are also deleted. You must have *SECADM or *ALLOBJ special authority to specify CAL(*ALL *ALL).

Some Ways to Use This Command

To delete your scheduling calendar, type:

```
DLTCAL CAL(*SCHED)
```

To delete all the calendars that you own, type:

```
DLTCAL CAL(*ALL)
```

To delete all the calendars that are in the calendar group called DEPT, type:

```
DLTCAL CAL(*GRP) GRPCALS(DEPT)
```

The group is not deleted, but it is empty.

To delete a specific calendar, PAT, for user PAT, type:

```
DLTCAL CAL(PAT PAT ROCH)
```

Note: The calendar name must be specified.

To delete all the calendars that are owned by all the users listed in a distribution list called DEPT, type:

```
DLTCAL CAL(*ALL DEPT DISTL)
```

Rules

Users cannot use calendar services when DLTCAL CAL(*ALL *ALL) is specified. This clears the database files.

Display Calendar Authority (DSPCALAUT) Command

The Display Calendar Authority (DSPCALAUT) command allows you to display the public and private authorities for a calendar. The authorities can be displayed on the screen, printed, or sent to a database file.

Security Considerations

If you are the owner of the calendar or have *SECADM or *ALLOBJ special authority, you see the *PUBLIC authority to the calendar and all the private authorities. Otherwise, you see only your private authority and the *PUBLIC authority to the calendar. If no private authorities are shown, the user has public authority to the calendar.

Some Ways to Use This Command

To display the authority information for your scheduling calendar, type:

```
DSPCALAUT
```

To print the authority information for the scheduling calendar of user PAT, type:

```
DSPCALAUT CAL(*SCHED PAT ROCH) OUTPUT(*PRINT)
```

To generate a database file containing the authority information for calendar CONFRM1, type:

```
DSPCALAUT CAL(CONFRM1 SUE ROCH) OUTPUT(*OUTFILE)  
OUTFILE(RMSLIB/CALAUT) OUTMBR(*FIRST *REPLACE)
```

Display Calendar Description (DSPCALD) Command

The Display Calendar Description (DSPCALD) command allows you to display information about one or more calendars. This information includes the manager of the calendar, the working times of the calendar, the owner, description, public and private authorities, and the notification information for entries on the calendar. The information can be displayed on the screen, printed, or sent to a database file.

Security Considerations

If you have *EXCLUDE authority to the calendar, you see only the calendar name, owner, description, and the assigned authorities.

If you have *TIMES authority to the calendar, you see the same information as a user with *EXCLUDE authority with the addition of the working times.

If you have *USE or greater authority, *SECADM or *ALLOBJ special authority, or you are the owner of the calendar, you see all information.

If you are displaying the calendar, the authorities shown depend on the security considerations described in the “Display Calendar Authority (DSPCALAUT) Command.”

If you are printing or generating an output file for the calendar, the authorities given are the public authorities and your current authorities to the calendar. Your authority is the same as the public authorities if you do not have a personal authority record for that calendar.

Some Ways to Use This Command

To display the calendar information for your scheduling calendar, type:

```
DSPCALD
```

To print the calendar information for the scheduling calendar of user PAT, type:

```
DSPCALD CAL(*SCHED PAT ROCH) OUTPUT(*PRINT)
```

To generate a database file containing the calendar information for calendar CONFRM1, type:

```
DSPCALD CAL(CONFRM1 SUE ROCH) OUTPUT(*OUTFILE) OUTFILE(RMSLIB/CAL)  
OUTMBR(*FIRST *REPLACE)
```

Display Calendar Item (DSPCALITM) Command

The Display Calendar Item (DSPCALITM) command allows you to display information about a calendar item. The information can be displayed on the screen, printed, or sent to a database file.

Security Considerations

If you have only *TIMES authority for an item, you see only the calendar name, owner, item ID, item type, date, and time. All the other information is blanked out.

Some Ways to Use This Command

The item ID values must be retrieved from output files from the QRYCALITM command or the ADDCALITM command. To display the calendar item that is in the CL variable &ITEM from your scheduling calendar, type the following line in a CL program:

```
DSPCALITM ITMID(&ITEM)
```

To print the calendar item that is in the CL variable &ITEM from the scheduling calendar of user PAT, type the following line in a CL program:

```
DSPCALITM ITMID(&ITEM) CAL(*SCHED PAT ROCH) OUTPUT(*PRINT)
```

To generate a database file containing the calendar item in the CL variable &ITEM from calendar CONFRM1, type the following line in a CL program:

```
DSPCALITM ITMID(&ITEM) CAL(CONFRM1 SUE ROCH) OUTPUT(*OUTFILE)  
OUTFILE(RMSLIB/CAL) OUTMBR(*FIRST *REPLACE)
```

End Calendar Service (ENDCALSRV) Command

The End Calendar Service (ENDCALSRV) command allows you to end the calendar services program. The command stops the job QCALSRV in a controlled manner with a delay time of 30 seconds to allow time for cleanup. The ENDCALSRV command retrieves job information (such as the job number, the user, and the job start date and time) so it can end the job.

Print Calendar (PRTCAL) Command

The Print Calendar (PRTCAL) command allows you to print one or more calendars. You can print calendars in a weekly (column) view, in a daily (list) view, in both views, or in a monthly view. You also can print groups of calendars in a column or a list view. The default format of the calendar printout is specified in your enroll-

ment record. Values specified on the PRTCAL command override your enrollment values.

Security Considerations

You must have at least *TIMES authority to print a calendar.

Some Ways to Use This Command

To print your scheduling calendar, for the items for the current job date, using all the defaults, type:

```
PRTCAL
```

To print two copies of the scheduling calendar for local user PAT, for the week containing March 1, 1990, in weekly format on printer PRT01, type:

```
PRTCAL CAL(*SCHED PAT ROCH) DATE(030190) PRTFMT(*COLUMNS)
DEV(PRT01) COPIES(2)
```

To print the calendars listed in the calendar group DEPT on a printout containing the calendars in a group format in 5 columns, type:

```
PRTCAL CAL(*GRP) GRPCALS(DEPT) CALTYP(*GRP) PRTFMT(*COLUMNS)
NBRCOLS(5)
```

Query Calendar Item (QRYCALITM) Command

The Query Calendar Item (QRYCALITM) command allows you to search one or more calendars for items that meet a given set of values. The values you can specify include the item type, item security, status, meeting information, job information, System/36 procedure information, text, date, day of week, and time.

The output from a search can be directed to a database file. You also receive a count of the number of items that match your search values, unless the specified time limit or selection limit is reached.

Some Ways to Use This Command

To search your scheduling calendar for all items between 8 a.m. and 10 a.m. on March 1, 1990, type:

```
QRYCALITM CAL(*SCHED) DATE(030190) TIME(8:00 10:00)
```

To search the scheduling calendars of all users in distribution list DEPT DISTL for events or meetings during March, type the following command. The items must be between 8 a.m. and 5 p.m. on Mondays, Wednesdays, or Fridays. The meetings must be tentative or confirmed, and the invitee's status must be unknown or attending. Matching items are written to output file QRYOUT in the first member of your library.

```
QRYCALITM CAL(*SCHED DEPT DISTL) ITMTYP(*EVENT *MEETING)
DATE(030190 033190) TIME ('8:00' '17:00') DAY(*MON *WED *FRI)
OUTFILE(*CURLIB/QRYOUT) MTGSTS(*TENTATIVE *CONFIRMED)
INVSTS(*ATTENDING *UNKNOWN)
```

To search all the calendars on the system for a scheduled job that is to use a SAVLIB command or a scheduled procedure that is to use a SAVLIBR procedure, type:

```
QRYCALITM CAL(*ALL *ALL) ITMTYP(*JOB *S36PRC) JOBCMD(SAVLIB)
S36PRC(SAVLIBR)
```

Rules

The following rules apply to the QRYCALITM command:

- You cannot query a calendar for items without *TIMES or greater authority. If authority is *TIMES, only the item type, date, day of week, and time values can be specified. Specifying any other values does not automatically return all items to which you have *TIMES authority. If output is directed to an output file, all items to which you have *TIMES authority and match the specified values have only partial data filled.
- Text searches for keywords cannot be split on two different text lines. For example, if the text keyword provided to query is 'dept meeting', then the entire keyword must appear on a single line within the item text. The word 'dept' cannot end on one line with 'meeting' starting on the next.

- Searches for job commands and System/36 procedure parameters are done the same as text searches. For example, if JOBCMD ('CRTLIB') is specified on query, then the following job command is considered a match.

```
SNDMSG MSG('Use the crtlib command')
```

Hints and Techniques

The following hints and techniques will help you use the QRYCALITM command:

- The best performance for the QRYCALITM command occurs when no text searches are specified. If text searches must be done, limit the number of keywords as much as possible and use the TEXTLGLREL(*OR) parameter when appropriate.
- If you want to check a calendar for free time, only items marked as 'busy' in the output file from the QRYCALITM command actually take up free time. For example, a reminder does not take any free time but a meeting with an invitee status of attending does take time.

Remove Calendar Item (RMVCALITM) Command

The Remove Calendar Item (RMVCALITM) command allows you to remove one or more items from one or more calendars. You can remove all items that are older than a certain date or between a given date range. You can also remove single items. The database files containing the calendar items can be reorganized after the records have been deleted. Reorganizing the database files recovers the space used by the deleted records, and improves the file performance.

Security Considerations

You must have *CHANGE or greater authority to remove any item from a calendar. You may remove items which you scheduled if you have *ADDITEM authority. A user with *SECADM or *ALLOBJ authority can remove any item, and is the only way CAL(*ALL *ALL) can be specified.

Some Ways to Use This Command

To remove the item that is contained in the CL variable &ITMID from your scheduling calendar, type the following in a CL program:

```
RMVCALITM CAL(*SCHED) ITMID(&ITMID)
```

To remove all items from all calendars you own that have a date before January 1, 1990, type:

```
RMVCALITM CAL(*ALL *CURRENT) DATE(*AVAIL 123189)
```

To remove all items that are older than 30 days from all calendars, type the following command. The space used by the deleted records is recovered.

```
RMVCALITM CAL(*ALL *ALL) ITMAGE(30) RGZPFM(*YES)
```

Rules

The following rules apply to the RMVCALITM command:

- When a meeting entry is removed, the meeting itself is not deleted. Only the entry on the specific calendar is removed. However, removing a meeting entry does change the user's invitee status to 'not attending'.
- When CAL(*ALL *ALL) is specified, any meetings in the given date range owned by a user with a calendar are deleted.

- If an item identifier (ITMID) is specified, the calendar (CAL) must specify a single calendar.

Restore Calendar (RSTCAL) Command

The Restore Calendar (RSTCAL) command allows you to restore calendars that were previously saved as part of a system backup of library QUSRSYS, or that were saved using the SAVCAL command. You may restore one or more calendars to the calendar they were saved from, or you may restore a calendar to a different calendar. If the calendar does not exist, it is created during the restore function.

Note: Calendars can be restored only from a system backup of the QUSRSYS library in Version 2 Release 1 Modification 0 or subsequent releases. A system backup of the QUSRSYS library from earlier releases cannot be used.

Security Considerations

When one or more saved calendars are being restored to the same calendar they were saved from, you must have *ADDITEM authority to the calendars. You can also restore previously saved calendars if you have *USE authority to the saved calendar, and *ADDITEM authority for the calendar to which the items are restored. A user with *SAVSYS special authority can restore any calendar to itself. A user with *SECADM or *ALLOBJ special authority can restore any calendar to any other calendar.

Some Ways to Use This Command

To restore your scheduling calendar from the device named TAP01, type:

```
RSTCAL CAL(*SCHED) DEV(TAP01)
```

To restore all the items dated after January 1, 1990 on all the calendars that the requester has *ADDITEM authority to, type the following command. The calendars are restored from the save file SAVEFILE1 in library MYLIB.

```
RSTCAL CAL(*ALL *ALL) Date(010190 *END) DEV(*SAVF) SAVF(MYLIB/SAVEFILE1)
```

To restore calendar MIKE, owned by user MIKE ROCH1, from diskette to calendar LAURIE, owned by user LAURIE ROCH2, type:

```
RSTCAL CAL(MIKE MIKE ROCH1) DEV(DKT01) TOCAL(LAURIE LAURIE ROCH2)
```

To restore the scheduling calendar for PAT ROCH from the system backup tape, type the following command:

```
RSTCAL CAL(*SCHED PAT ROCH) SAVLIB(QUSRSYS) LABEL(QUSRSYS) DEV(TAP01)
```

Rules

The following rules apply to the RSTCAL command:

- If the target calendar is not specified as TOCAL(*FROMCAL), the CAL parameter must specify a single calendar and the target calendar must already exist on the system. If the target calendar is specified as *FROMCAL, the CAL parameter can specify multiple calendars and the calendars are created if they do not exist (for example, generic *ALL).
- If a job or procedure is restored as a past date item, the job or procedure does not run.

- If an item is restored as a past date item, the reminder message indicator is always set to *NO.
- When calendars are being restored to the same calendar they were saved from, the scheduler of the item is not changed. If the original scheduler is no longer in the system distribution directory, the owner of the calendar becomes the scheduler of the items. When a calendar is restored to a different calendar, the user performing the restore becomes the scheduler of the items. Again, if the user performing the restore is not in the system distribution directory, the owner of the target calendar becomes the scheduler of the items.
- When restoring a calendar from a system backup, you must specify SAVLIB(QUSRSYS). The LABEL parameter must also be specified, for example, LABEL(QUSRSYS).

Hints and Techniques

The RSTCAL command is different from other system restore commands in that the calendar items are not replaced, they are merged into the target calendar as new items. Therefore, you may want to delete existing items before restoring items to a calendar to avoid duplicates.

The default LABEL parameter value of *GEN works only if you are restoring the same calendar that was saved using the SAVCAL command. If you are restoring a calendar saved using a different command, from a system backup, or saved prior to Version 2 Release 1 Modification 0, you must specify the label as it appears on the tape or diskette.

Calendars from all previous releases that were saved using the SAVCAL command or the interactive office save (used prior to Version 2 Release 1 Modification 0) can be restored. However, prior to Version 2 Release 1 Modification 0, only the calendar items were saved; thus, the target calendar must exist before restoring the calendar. In addition, only single calendars could be saved prior to Version 2 Release 1 Modification 0, so only a single calendar can be restored from these releases.

Save Calendar (SAVCAL) Command

The Save Calendar (SAVCAL) command allows you to save one or more calendars to tape, diskette, or a save file for backup or migration purposes.

Security Considerations

To save a calendar, you must have *USE authority for all security levels on that calendar. A user with *SAVSYS, *SECADM, or *ALLOBJ special authority can save any calendar.

Note: You may not always be able to restore a calendar just because you are able to save it. See the security considerations in “Restore Calendar (RSTCAL) Command” on page 2-16 for more information.

Some Ways to Use This Command

To save your scheduling calendar to the save file named SAVEFILE1 in library MYLIB, type:

```
SAVCAL CAL(*SCHED) DEV(*SAVF) SAVF(MYLIB/SAVEFILE1)
```

To save items dated before January 1, 1990, from all the calendars the requester has at least *USE authority to, type the following command. After the save, all items that the requester is authorized to remove, are removed. The calendars are saved to tapes TAP01 and TAP02.

```
SAVCAL CAL(*ALL *ALL) DATE(*AVAIL 123189) RMVITM(*YES)
      DEV(TAP01 TAP02)
```

To save the scheduling calendars for the users listed in the DEPT DSTL distribution list that the requester has *USE authority to, type the following command. The calendars are saved to diskette.

Note: The distribution list is not saved, it only serves as a way to specify at one time multiple calendars to save.

```
SAVCAL CAL(*SCHED DEPT DSTL) DEV(DKT01)
```

To save the scheduling calendar for user PAT ROCH so that it can be restored on a previous release system, type the following command:

```
SAVCAL CAL(*SCHED PAT ROCH) DEV(DKT01) TGTRLS(*PRV)
```

Rules

If you specified to remove items after they are saved, the items are not always removed. This is because more authority is needed to remove an item than is needed to save an item. However, an item is never removed unless it was first saved.

Hints and Techniques

You can quickly save all calendars on the system by specifying to save all calendars for the full date range. To do this you must have *SAVSYS, *SECADM, or *ALLOBJ special authority. If you do this, the calendar files are saved directly from QUSRSYS. Then, during the RSTCAL function, the saved library must be specified as QUSRSYS. No other users can be accessing calendar services for this save to work.

Start Calendar Service (STRCALSRV) Command

The Start Calendar Service (STRCALSRV) command allows you to start the calendar services program that notifies users about upcoming appointments and processes jobs scheduled on a calendar. The STRCALSRV command activates the job QCALSRV, which is placed in the QSYSNOMAX job queue and runs in the QSYSWRK subsystem.

When the calendar services program submits jobs to be run, it makes an entry in the system history log (QHST message queue).

Some Ways to Use This Command

You can start the calendar services program interactively by typing the following command:

```
STRCALSRV
```


Notes:

1. The library list used for a job is the same as the library list in the job description specified for the job.
2. The library list used when a procedure runs is the same as the library list for the active QCALSRV job. If you schedule a procedure that requires a specific library list, schedule it on the calendar using the Submit Job (SBMJOB) command.

You can add the STRCALSRV command to the system startup program.

Rules

Only one calendar service job can be active on the system at any time.

Calendar User-Defined Functions

Calendar user-defined functions allow users to use their applications from within the interactive calendar product. Each calendar user-defined function allows a user to define a short character string and associate any CL command with that string. When you type the function name, the CL command is run.

Information about the calendar and calendar items can be passed as parameters to the CL command. This allows you to use calendar-related information in other applications.

You can also receive information back from the command run by the calendar user-defined function. This information can be used to change the calendar view displayed before the calendar user-defined function was run. See “Control Calendar API (QOCTLCL)” on page 2-28 for more information on this function.

Security

Calendar user-defined functions can be owned by a specific user or *PUBLIC. Functions owned by *PUBLIC are considered public calendar user-defined functions and can be run by all users. Functions owned by a specific user are that user's **private** calendar user-defined functions. Only the owner of a private calendar user-defined function has the authority to use that function.

A security administrator with *SECADM authority can create public calendar user-defined functions for all calendar users and private calendar user-defined functions for specific users. Administrators also can change, copy, delete, and display all public and private calendar user-defined functions.

Users can create, change, copy, delete, and display their own private calendar user-defined functions. However, they can only display and print public calendar user-defined functions.

A user can have a private calendar user-defined function with the same name as a public calendar user-defined function. When the user uses the function, the private calendar user-defined function is run.

When a calendar user-defined function is run, the system checks to determine if the user has at least *USE authority to the command associated with the function. If the user does not have the proper authority, the command is not run.

Function Names

Calendar user-defined function names can be from 1 to 10 characters long. The first character must be a period (.) to distinguish the name from IBM-supplied function code names.

Valid names consist of any characters except the following: 0 through 9, asterisk (*), apostrophe ('), quotation marks ("), period (.), or a blank. A period (.) is always the first character in a name, but is not allowed anywhere else. Lower-case characters are mapped to their uppercase equivalent when entered (fields are only defined as uppercase).

If a calendar user-defined function refers to a displayed item on the calendar, the substitution variable &ITMID (item identifier) must be used in the command associated with the calendar user-defined function. If the item identifier is present, the calendar user-defined function name cannot be longer than 7 characters. This allows enough space for the item reference number, which has a range of 1 through 999.

Substitution Variables

Substitution variables are used to pass information from the current calendar display to the CL command run by the calendar user-defined function. All substitution variables begin with an ampersand (&).

Before running a calendar user-defined function, the command associated with the function is checked for any substitution variables. If a substitution variable is found, it is replaced with its associated value from the calendar display or the item identified by the item reference number.

The following substitution variables can be used in a calendar user-defined function:

| | |
|---------------------|---|
| &CALNAM | Calendar name that is currently being worked with. This value is set from the daily, weekly, reminder, monthly, and six-month calendar views. |
| &USRID | User ID that owns the calendar specified in &CALNAM. |
| &USRADR | Address associated with the user ID specified in &CALNAM. |
| &GRPNAM | Group name that is currently being worked with. This is set only if the user is on a multiple calendar view and working with a group calendar. |
| &DSTNAM | Distribution list identifier that is currently being worked with. This is set only if the user is on a multiple calendar view and working with a distribution list. |
| &DSTQUAL | List qualifier associated with the distribution list identifier. This is set only when &DSTNAM is set. |
| &CURDAT | Current date that the user is on in the calendar. The date is in the CYYMMDD format. |
| &CURDATJ | Current date that the user is on in the calendar. If the user chooses a range of dates, then the current date changes to the closest date to the one the user is currently working with. The date is in the current job format. |

| | | |
|--|------------------|---|
| | Composite | Today's date, unless the user chooses a different date. If the user scrolled to another week, then the current date is the first date in the farthest left column. |
| | Daily | First date displayed |
| | Group | Date shown |
| | Reminder | First date displayed |
| | Six month | First day of the first month shown |
| | Weekly | Today's date, unless the user chooses a different date. If the user scrolls to another week, then the current date is the first date in the farthest left column. |
| | Monthly | Today's date, unless the user chooses a different date. If the user scrolls to another week, then the current date is interpreted as the first date in the first row of the farthest left column. |

&STRDAT Start date that the calendar is on. The date is in the CYYMMDD format.

&STRDATJ Start date that the calendar is on. The date is in the current job format. Start dates are set for each of the views as follows:

| | |
|------------------|---|
| Composite | Date represented by the farthest left column on the display |
| Daily | First date displayed |
| Group | Date shown (same as current date) |
| Reminder | First date displayed |
| Six month | First day of the first month shown |
| Weekly | Date represented by the farthest left column on the display |
| Monthly | First day of the current month shown |

&ENDDAT End date that the calendar is on. The date is in the CYYMMDD format.

&ENDDATJ End date that the calendar is on. The date is shown in the current job format. End dates are set for each of the views as follows:

| | |
|------------------|--|
| Composite | Date represented by the farthest right column on the display. |
| Daily | Last date shown. This is the last date the user sees if the Page Down key is held down, not the last date that is on the current screen of data. |
| Group | Date shown (same as start date and current date). |
| Reminder | Last date shown. This is the last date the user sees if the Page Down key is held down, not the last date that is on the current screen of data. |

| | | |
|----------------------|------------------|--|
| | Six month | Last day on the last month shown. |
| | Weekly | Date represented by the farthest right column on the display. |
| | Monthly | Last date of the current month on the display. |
| &CURDAY | | Day represented by the date given in the current date. Possible values are as follows: |
| | *SUN | Sunday |
| | *MON | Monday |
| | *TUE | Tuesday |
| | *WED | Wednesday |
| | *THU | Thursday |
| | *FRI | Friday |
| | *SAT | Saturday |
| &STRDAY | | Day represented by the date given in the start date (substitution variable &STRDAT or &STRDATJ). Possible values are as follows: |
| | *SUN | Sunday |
| | *MON | Monday |
| | *TUE | Tuesday |
| | *WED | Wednesday |
| | *THU | Thursday |
| | *FRI | Friday |
| | *SAT | Saturday |
| &ENDDAY | | Day represented by the date given in the end date (substitution variable &ENDDAT or &ENDDATJ). Possible values are as follows: |
| | *SUN | Sunday |
| | *MON | Monday |
| | *TUE | Tuesday |
| | *WED | Wednesday |
| | *THU | Thursday |
| | *FRI | Friday |
| | *SAT | Saturday |
| &CURWKNBR | | A four byte variable containing the current week and year, in the format of WWYY. If the current date and week number are both supplied, the week number is ignored. |
| &STRWKNBR | | A four byte variable containing the starting week and year, in the format of WWYY. |
| &ENDWKNBR | | A four byte variable containing the ending week and year, in the format of WWYY. |
| &FNCNAM | | User-defined function that was typed. If this function allows an item number (such as .PRnn), only the function name is passed (such as .PR). |
| &FCNTXT | | Text associated with the user-defined function. |

&ITMID The 20-byte item identifier that the user selected. For example, if the user specified .PR6, this substitution variable is replaced with the item identifier associated with item 6 on the current view of the calendar.

&GRPUSRID User ID that owns the calendar specified in &GRPNAM.

&GRPUSRADR Address associated with the user ID specified in &GRPUSRID.

&CALVIEW View of the calendar that the calendar user-defined function was entered on. Possible values are as follows:

| | |
|-------------------|--------------------------------|
| *WEEKLY | Weekly view of calendar |
| *DAILY | Daily view of calendar |
| *GROUP | Group view of calendar |
| *MONTHLY | Monthly view of calendar |
| *COMPOSITE | Composite view of calendar |
| *REMINDER | Reminder-only view of calendar |
| *SIXMONTH | Six-month view of calendar |

The following substitution variables are converted to the current job CCSID before being set in the command:

| | | |
|---------|----------|------------|
| &CALNAM | &DSTNAM | &FCNTXT |
| &USERID | &DSTQUAL | &GRPUSRID |
| &USRADR | &FCNNAM | &GRPUSRADR |
| &GRPNAM | | |

Examples

The following examples show how you can use calendar user-defined functions.

Example 1

This example shows a calendar user-defined function that allows you to work with your submitted jobs. When you type `.WSJ` at the function line on a main calendar display, the Work with Submitted Jobs (WRKSBMJOB) command runs and displays the Work with Submitted Jobs display. The following shows the Create User-Defined Function display where you type the function name.

Create User-Defined Function

Type choices, press Enter.

| | | |
|--------------------|----------------------------------|------|
| Function | .WSJ_____ | Name |
| Command | WRKSBMJOB_____ | |
| _____ | | |
| _____ | | |
| _____ | | |
| F4 for prompt | | |
| Text | Work with my submitted jobs_____ | |

F3=Exit F4=Prompt F5=Refresh F9=Six month calendar F12=Cancel
F19=Display messages

Example 2

This example shows a CL program that uses information passed to it from the main calendar display. The CL program displays the Query Calendar Item (QRYCALITM) display for you to enter the selection values. After the selection values are made and you press the Enter key, the Display Calendar Item (DSPCALITM) display appears for each item that was selected. The calendar name, start date, and end date parameters for the QRYCALITM command are provided from the main calendar display.

Type the calendar user-defined function name (.DQI) on the Create User-Defined Function display.

Create User-Defined Function

Type choices, press Enter.

FunctionDQI_____ Name

Command CALL PGM(*LIBL/DSPQRYITM) PARM('&CALNAM &U_____ SRID &USRADR &GRPNAM &GRPUSRID &GRPUSRADR &DSTNAM &DSTQUAL &STRDATJ &END_____ DATJ')_____

F4 for prompt

Text Display queried calendar items_____

F3=Exit F4=Prompt F5=Refresh F9=Six month calendar F12=Cancel
 F19=Display messages

The following shows the source code for the sample CL program, DSPQRYITM:

```

/**** START OF SPECIFICATIONS *****/
/*                                                                    */
/* MODULE NAME: DSPQRYITM                                             */
/*                                                                    */
/* DESCRIPTIVE NAME: Display calendar items matching a query         */
/*                                                                    */
/* FUNCTION/PURPOSE:                                                 */
/* Prompt QRYCALITM to find all items matching a given set of       */
/* criteria and then display those items to the screen.              */
/*                                                                    */
/* DEPENDENCIES:                                                     */
/* QRYCALITM and DSPCALITM CL commands (V2R1 0V/400).               */
/*                                                                    */
/* INPUT:                                                            */
/* CALNAM -The 10-byte calendar name                                  */
/* USRID -The 8-byte user id that owns the calendar                  */
/* USRADR -The 8-byte user address that owns the calendar            */
/* GRPNAM -The 10-byte group name                                     */
/* GRPUSRID -The 8-byte user id that owns the group calendar        */
/* GRPUSRADR-The 8-byte user address that owns the group calendar*/
/* DSTNAM -The 8-byte distribution list name                          */
/* DSTQUAL -The 8-byte distribution list qualifier                   */
/* STRDATJ -The 9-byte start date of the calendar in job format     */
/* ENDDATJ -The 9-byte end date of the calendar in job format      */
/*                                                                    */
/*****/

PGM            PARM(&CALNAM &USRID &USRADR &GRPNAM +
                 &GRPUSRID &GRPUSRADR &DSTNAM &DSTQUAL +
                 &STRDATJ &ENDDATJ)

```

Figure 2-1 (Part 1 of 3). Source Code for DSPQRYITM CL Program

```

/*                                                                 */
/* Declare parameters                                             */
/*                                                                 */
      DCL          VAR(&CALNAM) TYPE(*CHAR) LEN(10)
      DCL          VAR(&USRID) TYPE(*CHAR) LEN(8)
      DCL          VAR(&USRADR) TYPE(*CHAR) LEN(8)
      DCL          VAR(&GRPNAM) TYPE(*CHAR) LEN(10)
      DCL          VAR(&GRPUSRID) TYPE(*CHAR) LEN(8)
      DCL          VAR(&GRPUSRADR) TYPE(*CHAR) LEN(8)
      DCL          VAR(&DSTNAM) TYPE(*CHAR) LEN(8)
      DCL          VAR(&DSTQUAL) TYPE(*CHAR) LEN(8)
      DCL          VAR(&STRDATJ) TYPE(*CHAR) LEN(9)
      DCL          VAR(&ENDDATJ) TYPE(*CHAR) LEN(9)
/*                                                                 */
/* Declare local variables                                       */
/*                                                                 */
      DCLF         FILE(QOFC/QAOYQRYITM)
      DCL          VAR(&CALNAME) TYPE(*CHAR) LEN(10)
      DCL          VAR(&USERID) TYPE(*CHAR) LEN(8)
      DCL          VAR(&ADDRESS) TYPE(*CHAR) LEN(8)
      DCL          VAR(&CHARSET) TYPE(*CHAR) LEN(4)
      DCL          VAR(&CODEPAGE) TYPE(*CHAR) LEN(4)
      DCL          VAR(&ITMID) TYPE(*CHAR) LEN(20)
/*                                                                 */
/* Do a QRYCALITM to outfile member QTEMP/QRYCALITM            */
/*                                                                 */
      CLRPFM       FILE(QTEMP/QRYCALITM) MBR(QRYCALITM)
      MONMSG       MSGID(CPF0000 OFC0000)
/*                                                                 */
/* Current calendar view is a calendar                          */
/*                                                                 */
      IF          COND((&CALNAM *NE ' ') *AND (&USRID *NE ' ') +
                      *AND (&USRADR *NE ' ')) THEN(DO)
      ?          QRYCALITM CAL(&CALNAM &USRID &USRADR) +
                DATE((&STRDATJ &ENDDATJ)) +
                OUTFILE(QTEMP/QRYCALITM) OUTMBR(QRYCALITM)
      MONMSG      MSGID(0FC312D) EXEC(GOTO CMDLBL(CONTINUE))
      MONMSG      MSGID(CPF0000 OFC0000) EXEC(GOTO +
                CMDLBL(QRYERR))
      ENDDO
/*                                                                 */
/* Current calendar view is a group calendar                    */
/*                                                                 */
      IF          COND((&GRPNAM *NE ' ') *AND (&GRPUSRID *NE ' ' +
                      ') *AND (&GRPUSRADR *NE ' ')) THEN(DO)
      ?          QRYCALITM CAL(*GRP) GRPCALS(&GRPNAM +
                &GRPUSRID &GRPUSRADR) DATE((&STRDATJ +
                &ENDDATJ)) OUTFILE(QTEMP/QRYCALITM) +
                OUTMBR(QRYCALITM)
      MONMSG      MSGID(0FC312D) EXEC(GOTO CMDLBL(CONTINUE))
      MONMSG      MSGID(CPF0000 OFC0000) EXEC(GOTO +
                CMDLBL(QRYERR))
      ENDDO

```

Figure 2-1 (Part 2 of 3). Source Code for DSPQRYITM CL Program


```

/*                                                                    */
/* Current calendar view is a distribution list                        */
/*                                                                    */
      IF          COND((&DSTNAM *NE ' ') *AND (&DSTQUAL *NE ' ' +
          ?          QRYCALITM CAL(*ALL &DSTNAM &DSTQUAL) +
                    DATE((&STRDATJ &ENDDATJ)) +
                    OUTFILE(QTEMP/QRYCALITM) OUTMBR(QRYCALITM)
MONMSG          MSGID(OFC312D) EXEC(GOTO CMDLBL(CONTINUE))
MONMSG          MSGID(CPF0000 OFC0000) EXEC(GOTO +
          CMDLBL(QRYERR))
      ENDDO

/*                                                                    */
/* Loop through QRYCALITM outfile and get details for each item     */
/* Then display the item                                             */
/* Note: variables QCIID, QCCAL, QCODEN, QCODGN are defined         */
/* in the DDS for QAOYQRYITM                                        */
/*                                                                    */
CONTINUE:      OVRDBF          FILE(QAOYQRYITM) TOFILE(QTEMP/QRYCALITM) +
          MBR(QRYCALITM)

LOOP:          RCVF
MONMSG          MSGID(CPF0864) EXEC(GOTO CMDLBL(END))
CHGVAR          VAR(&ITMID) VALUE(&QCIID)
CHGVAR          VAR(&CALNAME) VALUE(&QCCAL)
CHGVAR          VAR(&USERID) VALUE(&QCODEN)
CHGVAR          VAR(&ADDRESS) VALUE(&QCODGN)
CHGVAR          VAR(&CHARSET) VALUE(&QCCALC)
CHGVAR          VAR(&CODEPAGE) VALUE(&QCCALP)
DSPCALITM      ITMID(&ITMID) CAL(&CALNAME &USERID &ADDRESS) +
          OUTPUT(*) CMDCHRID(&CHARSET &CODEPAGE)
MONMSG          MSGID(CPF0000 OFC0000) EXEC(GOTO +
          CMDLBL(DSPERR))
      GOTO          CMDLBL(LOOP)

/*                                                                    */
/* Signal fatal error sent by QRYCALITM and quit                    */
/*                                                                    */
QRYERR:        SNDPGMMSG      MSG('Fatal error on QRYCALITM command. See +
          previous messages.')
      GOTO          CMDLBL(END)

/*                                                                    */
/* Signal error sent by DSPCALITM and quit                          */
/*                                                                    */
DSPERR:        SNDPGMMSG      MSG('Fatal error on DSPCALITM command. See +
          previous messages.')
      GOTO          CMDLBL(END)

/*                                                                    */
/* Signal any necessary errors and quit                              */
/*                                                                    */
END:          ENDPGM

```

Figure 2-1 (Part 3 of 3). Source Code for DSPQRYITM CL Program

Control Calendar API (QOOCTLCL)

The Control Calendar application program interface (API), QOOCTLCL, allows application programs to control how a calendar is displayed on returning from a calendar user-defined function or when it is initially displayed.

You can use the QOOCTLCL API in two ways. The program run by the command associated with a calendar user-defined function can call the QOOCTLCL API to change the current calendar view when it returns to calendar services. Or, you can call the QOOCTLCL API before entering calendar services to change the current calendar view without changing your calendar user profile information.

The QOOCTLCL API has these parameters:

| QOOCTLCL: Required Parameters | | | |
|-------------------------------|--|--------------|-----------------|
| 1 | Calendar view type | INPUT | CHAR(10) |
| | <p>The type of view that should be used when the calendar is displayed. Use one of the following special values:</p> <p>*WEEKLY Weekly calendar view. Allowed for single calendars only. If this value is specified and the calendar parameter and distribution parameters are both blank, the view is ignored if it is not compatible with the current or default calendar being displayed.</p> <p>*DAILY Daily calendar view. Allowed for single and multiple calendars.</p> <p>*GROUP Group calendar view. Allowed for multiple calendars only. If this value is specified and the calendar parameter and distribution parameters are both blank, the view is ignored if it is not compatible with the current or default calendar being displayed.</p> <p>*COMPOSITE Composite calendar view. Allowed for multiple calendars only. If this value is specified and the calendar parameter and distribution parameters are both blank, the view is ignored if it is not compatible with the current or default calendar being displayed.</p> <p>*MONTHLY Monthly calendar view. Allowed for single or multiple calendars.</p> <p>*SIXMONTH Six month calendar view. Allowed for single and multiple calendars.</p> <p>*REMINDER Reminder only calendar view. Allowed for single and multiple calendars.</p> <p>*NONE Do not control the calendar view. Either the current calendar view or the default view for the single or multiple calendar specified is used.</p> | | |

| QOOCTLCL: Required Parameters | | | |
|--------------------------------------|---|--------------|----------------|
| 2 | Current date | INPUT | CHAR(7) |
| | <p>The current date that should be used when the calendar is displayed. The date must be specified in the format CYYMMDD, where</p> <p>C = century (0=1900, 1=2000) YY = year MM = month DD = day</p> <p>You can use the following special value: *NONE Do not control the current date.</p> | | |
| 3 | Start time | INPUT | CHAR(6) |
| | <p>The start time that should be used when the calendar is displayed. The time must be in 24-hour format in the format HHMMSS, where</p> <p>HH = hours MM = minutes SS = seconds</p> <p>The seconds must be a valid value (00–59), but is ignored. You can use the following special value: *NONE Do not control the start time.</p> | | |
| 4 | Number of columns | INPUT | BIN(4) |
| | <p>The number of columns that should be used when the calendar is displayed. The correct values are 5, 6, or 7. If the view used when the calendar is displayed is not one of the columnar displays, this value is ignored. If the number of columns specified is 6 or 7 and the current work station does not support more than 5 columns, this value is ignored. You can use the following special value: 0 Do not control the number of columns.</p> | | |
| 5 | Time interval | INPUT | BIN(4) |
| | <p>The time interval that should be used when the calendar is displayed. The correct values for the calendar view types *WEEKLY, *GROUP, and *COMPOSITE, are 5 through 60. The correct values for calendar view type *MONTHLY are 5 through 120. When the calendar is displayed, the interval used is adjusted to the nearest even divisor of 60, 90, or 120. This value is ignored if the view used for the calendar is not one of the columnar displays. You can use the following special value: 0 Do not control the time interval.</p> | | |
| 6 | Calendar | INPUT | CHAR(*) |
| | <p>A list of calendar names and owners that should be used when the calendar is displayed. Each element in the list is 26 characters in length. The first 10 characters are the calendar name, the next 8 characters are the owner's user ID, and the last 8 characters are the owner's address. The number of elements in the list should equal the value for the number of calendars parameter. All characters for this parameter should be in the current job's CCSID value.</p> | | |

| QOOCTLCL: Required Parameters | | | |
|-------------------------------|---|--------------|-------------------|
| 7 | Number of calendars | INPUT | BIN(4) |
| | <p>The number of calendars that are specified on the calendar parameter.</p> <p>This parameter should be 0 if the calendar parameter does not specify any calendars. This parameter must be 0 if either the group calendar or distribution list parameter is not blank.</p> | | |
| 8 | Group calendar | INPUT | CHAR(26) |
| | <p>The name of a group calendar and owner that should be used when the calendar is displayed. The first 10 characters are the group calendar name, the next 8 characters are the owner's user ID, and the last 8 characters are the owner's address. All characters for this parameter should be in the current job's CCSID value.</p> <p>This parameter should be all blanks if a group calendar should not be shown when the calendar is displayed. This parameter must be blank if the distribution list parameter is not blank or if the number of calendars parameter is greater than 0.</p> | | |
| 9 | Distribution list | INPUT | CHAR(16) |
| | <p>The name of a distribution list that should be used when the calendar is displayed. The first 8 characters are the distribution list name and the last 8 characters are the distribution list qualifier. All characters for this parameter should be in the current job's CCSID value.</p> <p>This parameter should be all blanks if a distribution list should not be shown when the calendar is displayed. This parameter must be blank if the group calendar parameter is not blank or if the number of calendars parameter is greater than 0.</p> | | |
| 10 | Error Code | I/O | ERROR CODE |
| | <p>For complete information on the error code parameter, see the <i>System Programmer's Interface Reference</i> manual.</p> | | |
| 11 | Current week | INPUT | CHAR(5) |
| | <p>The current week that should be used when the calendar is displayed. The week must be specified as WWYY, where WW is the week number and YY is the year. You can use the following special value:</p> <p>*NONE Do not control the current week.</p> <p>If both the current day and current week parameters are specified, the current week number is ignored and the current date value is used.</p> | | |

Control Calendar API Example

This section provides an example of how you can use the QOOCTLCL API. The example shows a CL program that uses information passed to it from the main calendar display. Information is then returned back to the calendar. The CL program displays the Query Calendar Item (QRYCALITM) display so you can type the selection criteria. The items selected are then written to an outfile. The outfile is read and displayed, one page at a time, by the RPG program shown in Figure 2-3 on page 2-35. You can then select an item to be displayed and how the calendar is to display the item on return.

Type the calendar user-defined function .SQL on the Create User-Defined Function display to call the SLTQRYITM CL program.


```

/**** START OF SPECIFICATIONS *****/
/*
/* PROCESSOR: CL PROGRAM
/*
/* MODULE NAME: SLTQRYITM
/*
/* DESCRIPTIVE NAME: Select items to display matching query
/*
/* FUNCTION/PURPOSE:
/* Display the Queried Calendar Items panel and process the
/* users input.
/*
/* DEPENDENCIES:
/* QRYCALITM CL commands (V2R1 OV/400).
/*
/* EXTERNAL REFERENCES:
/* QRYCALIT RPG program
/*
/* INPUT:
/* CALNAM -The 10-byte calendar name
/* USRID -The 8-byte user id that owns the calendar
/* USRADR -The 8-byte user address that owns the calendar
/* GRPNAM -The 10-byte group name
/* GRPUSRID -The 8-byte user id that owns the group calendar
/* GRPUSRADR-The 8-byte user address that owns the group calendar*/
/* DSTNAM -The 8-byte distribution list name
/* DSTQUAL -The 8-byte distribution list qualifier
/* STRDATJ -The 9-byte start date of the calendar in job format
/* ENDDATJ -The 9-byte end date of the calendar in job format
/*
/* OUTPUT:
/* NONE
/*
/*****/
                PGM                PARM(&CALNAM &USRID &USRADR &GRPNAM +
                                &GRPUSRID &GRPUSRADR &DSTNAM &DSTQUAL)
/*
/* Declare parameters
/*
                DCL                VAR(&CALNAM) TYPE(*CHAR) LEN(10)
                DCL                VAR(&USRID) TYPE(*CHAR) LEN(8)
                DCL                VAR(&USRADR) TYPE(*CHAR) LEN(8)
                DCL                VAR(&GRPNAM) TYPE(*CHAR) LEN(10)
                DCL                VAR(&GRPUSRID) TYPE(*CHAR) LEN(8)
                DCL                VAR(&GRPUSRADR) TYPE(*CHAR) LEN(8)
                DCL                VAR(&DSTNAM) TYPE(*CHAR) LEN(8)
                DCL                VAR(&DSTQUAL) TYPE(*CHAR) LEN(8)
/*
/* Declare local variables
/*
                DCL                VAR(&CALNAME) TYPE(*CHAR) LEN(10)

```

Figure 2-2 (Part 1 of 3). Source Code for SLTQRYITM CL Program

```

/*                                                                    */
/* Do a QRYCALITM to outfile mbr QTEMP/QRYCALITM                      */
/*                                                                    */
        CLRPFM      FILE(QTEMP/QRYITMF) MBR(QRYITMF)
        MONMSG     MSGID(CPF0000)
        MONMSG     MSGID(CPF9999)
        MONMSG     MSGID(OFC0000)

/*                                                                    */
/* Current calendar view is a calendar                                */
/*                                                                    */
        IF          COND((&CALNAM *NE ' ') *AND (&USRID *NE ' ') +
        ?           *AND (&USRADR *NE ' ')) THEN(DO)
        QRYCALITM CAL(&CALNAM &USRID &USRADR) +
        OUTFILE(QTEMP/QRYITMF) OUTMBR(QRYITMF)
        MONMSG     MSGID(OFC312D) EXEC(GOTO CMDLBL(CONTINUE))
        MONMSG     MSGID(CPF0000 OFC0000) EXEC(GOTO +
        CMDLBL(QRYERR))
        GOTO       CMDLBL(CONTINUE)
        ENDDO

/*                                                                    */
/* Current calendar view is a group calendar                          */
/*                                                                    */
        IF          COND((&GRPNAM *NE ' ') *AND (&GRPUSRID *NE ' ' +
        ?           ') *AND (&GRPUSRADR *NE ' ')) THEN(DO)
        QRYCALITM CAL(*GRP) GRPCALS(&GRPNAM +
        &GRPUSRID &GRPUSRADR) +
        OUTFILE(QTEMP/QRYITMF) OUTMBR(QRYITMF)
        MONMSG     MSGID(OFC312D) EXEC(GOTO CMDLBL(CONTINUE))
        MONMSG     MSGID(CPF0000 OFC0000) EXEC(GOTO +
        CMDLBL(QRYERR))
        GOTO       CMDLBL(CONTINUE)
        ENDDO

/*                                                                    */
/* Current calendar view is a distribution list                       */
/*                                                                    */
        IF          COND((&DSTNAM *NE ' ') *AND (&DSTQUAL *NE ' ' +
        ?           ')) THEN(DO)
        QRYCALITM CAL(*ALL &DSTNAM &DSTQUAL) +
        OUTFILE(QTEMP/QRYITMF) OUTMBR(QRYITMF)
        MONMSG     MSGID(OFC312D) EXEC(GOTO CMDLBL(CONTINUE))
        MONMSG     MSGID(CPF0000 OFC0000) EXEC(GOTO +
        CMDLBL(QRYERR))
        GOTO       CMDLBL(CONTINUE)
        ENDDO

/*                                                                    */
/* Override the database file                                        */
/*                                                                    */
        CONTINUE:  OVRDBF      FILE(QA0YQRYITM) TOFILE(QTEMP/QRYITMF) +
        MBR(QRYITMF) SEQONLY(*NO)

/*                                                                    */
/* Call the RPG select queried items program                        */
/*                                                                    */
        CALL       PGM(*LIBL/SLTQRYIT)
        MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(SLTERR))
        GOTO       CMDLBL(END)

```

Figure 2-2 (Part 2 of 3). Source Code for SLTQRYITM CL Program

```

/* */
/* Signal error on QRYCALITM command and quit */
/* */
QRYERR:      SNDPGMSG MSG('Errors occurred on QRYCALITM command. +
                See previous messages.')
                GOTO      CMDLBL(END)

/* */
/* Signal error on SLTCALIT program and quit */
/* */
SLTERR:      SNDPGMSG MSG('Errors occurred on SLTQRYIT program. +
                See previous messages.')
                GOTO      CMDLBL(END)

/* */
/* Signal any necessary errors and quit */
/* */
END:         ENDPGM

```

Figure 2-2 (Part 3 of 3). Source Code for SLTQRYITM CL Program

Figure 2-3 shows the RPG program called by the SLTQRYITM CL program.

```

/*****      START OF SPECIFICATIONS      *****/
/*
/* MODULE NAME: SELECT QUERIED ITEMS
/*
/* DESCRIPTIVE NAME: SELECT QUERIED CALENDAR ITEMS TO DISPLAY
/*
/* FUNCTION: THIS PROGRAM WILL ALLOW THE USER TO DISPLAY THE
/*           QUERIED CALENDARS UPON RETURN TO CALENDAR SERVICES.
/*
/*
/* DEPENDENCIES:
/*   QOCTLCL - Control calendar API (V2R2 OV/400)
/*   QRYCALIT - CL program that generates the QRYITMF file
/*   SENDMSG - CL program that sends/removes messages to/from
/*             this programs message queue
/*
/* EXTERNAL REFERENCES:
/*   QOCTLCL - Control calendar API
/*   SENDMSG - Send/remove messages to/from this programs
/*             message queue
/*
/* INPUT:
/*
/* OUTPUT:
/*
/* EXIT NORMAL: RETURN TO CALLING PROGRAM
/*
/* EXIT ERROR:  FUNCTION CHECK
/*
/*
/*****      END OF SPECIFICATIONS      *****/
H
FQRYITMF IF E          DISK          KRECNO REC#
FDQRYITM CF E        WORKSTN       KINFDS FEEDBK
F                      SUB#        KSFIL# SUB
F*
E* Compile time arrays
E          ERR        1   5  50          ERROR MESSAGES
E          VTP        1   5  10          VIEW TYPES
E          ITM        1   5   9          ITEM TYPES
E          ITS        1   5   3          ITEM TYPES (Abrv)
E          NBR        1  10   1          NUMBER (0-9)
E*
E* Relative record number and option field arrays
E          RCD        50  4  0          RECORD NUMBERS
E          OPT        50  1           OPTIONS
E*
IFEEDBK    DS
I                      B 370 3710CLOC
ISUBDSP    DS
I                      1   3 PITEM
I                      4  11 PSDATE
I                      12  17 PSTIME
I                      12  13 PHOURS

```

Figure 2-3 (Part 1 of 17). RPG Program Called by SLTQRYITM CL Program

```

I          12 130ZHOURLS
I          18 27 PCALNM
I          28 35 PCALDN
I          36 43 PCALDG
I          44 63 PTEXT
I          64 670PRECNO
I* Current date - conversion to MM/DD/YY format
I          DS
I          1  8 STRDAT
I          1  2 MM
I          3  3 #SEP1
I          4  5 DD
I          6  6 #SEP2
I          7  8 YY
I* Current date - conversion to CYYMMDD format
I          DS
I          1  7 CURDAT
I          1  1 CENTRY
I          2  7 YYMMDD
I* Start time - conversion to HHMMSS format
I          DS
I          1  6 STRTIM
I          1  2 HOURS
I          1  1 HOUR1
I          2  2 HOUR2
I          3  4 MINS
I          5  6 SECS
I* 26 byte calendar name
I          DS
I          1 26 CALNDR
I          1 10 CALNAM
I          11 18 CALDEN
I          19 26 CALDGN
I* Conversion from character to zoned
I          DS
I          1  2 CINTVL
I          1  1 CINTV1
I          2  2 CINTV2
I          1 20ZINTVL
I* Conversion from character to zoned
I          DS
I          1  1 CCOLS
I          1 10ZCOLS
I* Binary values for call to Q00CTLCL API
I          DS
I          B  1  40COLMNS
I          B  5  80INTRVL
I          B  9 120CALNBR
I          B 13 160ERRCD
I*

```

Figure 2-3 (Part 2 of 17). RPG Program Called by SLTQRYITM CL Program

```

C*****
C/EJECT
C*
C*   INDICATOR      MEANING
C*   -----      -
C*   01             ** NOT USED **
C*   02             ** NOT USED **
C*   03             PF03 EXIT
C*   04             ** NOT USED **
C*   05             PF05 REFRESH
C*   06 - 29        ** NOT USED **
C*   30             WORK INDICATOR
C*   31             WORK INDICATOR
C*   32             WORK INDICATOR
C*   33 - 69        ** NOT USED **
C*   70 - 79        ** NOT USED **
C*   80             ** NOT USED **
C*   81             REVERSE IMAGE - VIEW TYPE
C*   82             REVERSE IMAGE - COLUMNS
C*   83             REVERSE IMAGE - INTERVAL
C*   84             REVERSE IMAGE - OPTION
C*   85 - 89        ** NOT USED **
C*   90 - 93        ** NOT USED **
C*   94             SFLEND FOR MESSAGE SUBFILE
C*   96             DISPLAY RECORD SUBFILE
C*   97             UNDERLINE OPTION FIELD
C*   99             CLEAR RECORD SUBFILE
C*
C/EJECT
C*****
C* INITIALIZATION
C*****
C*
C* Define constant *NONE
C           MOVE1*NONE   'NONE10 10
C*
C* Define constants for view type, columns, time interval,
C* and option field line and position
C           Z-ADD5       #VTPLN  20
C           Z-ADD37      #VTPPS  20
C           Z-ADD7       #CLSLN  20
C           Z-ADD37      #CLSPS  20
C           Z-ADD8       #INTLN  20
C           Z-ADD37      #INTPS  20
C           Z-ADD13      #OPTLN  20
C           Z-ADD3       #OPTPS  20
C* Define constant for page size and page size times 2
C           Z-ADD7       #PAGE   20
C           Z-ADD14      #PAGE2  20
C*

```

Figure 2-3 (Part 3 of 17). RPG Program Called by SLTQRYITM CL Program

```

C* Define constants for columns and time interval
C* maximum and minimum values
C          Z-ADD5          #MNCOL  20
C          Z-ADD7          #MXCOL  20
C          Z-ADD5          #MNINT  20
C          Z-ADD60         #MXINT  20
C*
C* Define constants for date separator, a.m., and p.m.
C          MOVE '/'        #SEP1
C          MOVE '/'        #SEP2
C          MOVE 'a'        #AM      1
C          MOVE 'p'        #PM      1
C*
C* Initialize the program queue
C          MOVE'L'SLTQRYIT'PGMQ
C*
C* Initialize the panel variables
C          MOVE *BLANK     PVTTYPE
C          MOVE *BLANK     PCOLS
C          MOVE *BLANK     PINTVL
C          Z-ADD0          LNNBR   30
C          Z-ADD0          PSNBR   30
C          MOVE 'Y'        REDSPY  1
C          MOVE 'N'        PNLERR  1
C          MOVE 'N'        ONEOPT  1
C          Z-ADD0          REC#    40
C          Z-ADD0          SUB#    40
C*
C/EJECT
C*****
C*          ***          START OF MAINLINE          ***
C*****
C*
C* Clear the record subfile
C          MOVE '1'        *IN99
C          WRITESUBCTL
C          MOVE '0'        *IN99
C*
C* Build the record subfile
C          Z-ADD1          REC#
C          EXSR BLDSUB
C*
C* Signal to display the record subfile
C          MOVE '1'        *IN96
C          MOVE '1'        *IN94
C*

```

Figure 2-3 (Part 4 of 17). RPG Program Called by SLTQRYITM CL Program

```

C*****
C* DISPLAY THE PANEL(S)
C*****
C*
C          REDSPY    DOWEQ 'Y'
C          MOVE 'N'      REDSPY
C*
C          SUB#      WRITEMSGCTL
C          IFEQ 0
C          WRITEPANEL3
C          ELSE
C          WRITEPANEL2
C          WRITESUBCTL
C          END
C*
C          EXFMTPANEL1
C*
C          SELEC
C*
C*****
C* WHEN F03 PRESSED, RETURN
C*****
C          *IN03    WHEQ '1'
C          SETON          LR
C          RETRN
C*
C*****
C* WHEN F05 PRESSED, REFRESH
C*****
C          *IN05    WHEQ '1'
C          EXSR REFRSH
C          MOVE 'Y'      REDSPY
C*
C*****
C* WHEN F07 PRESSED, ROLL UP
C*****
C          *IN07    WHEQ '1'
C          EXSR CHKERR
C*
C          PNLERR   IFEQ 'Y'
C          REC#     IFGT #PAGE
C          SUB #PAGE   REC#
C          ELSE
C          Z-ADD1     REC#
C          END
C          END
C*
C          EXSR BLDSUB
C          MOVE 'Y'      REDSPY
C*

```

Figure 2-3 (Part 5 of 17). RPG Program Called by SLTQRYITM CL Program

```

C*****
C* WHEN F08 PRESSED, ROLL DOWN
C*****
C          *IN08      WHEQ '1'
C                      EXSR CHKERR
C*
C          PNLERR     IFEQ 'Y'
C          REC#       IFGT #PAGE
C                      SUB #PAGE      REC#
C                      ELSE
C                      Z-ADD1          REC#
C                      END
C*
C                      ELSE
C          REC#       IFGT #PAGE2
C                      SUB #PAGE2     REC#
C                      ELSE
C                      Z-ADD1          REC#
C                      END
C                      END
C*
C                      EXSR BLDSUB
C                      MOVE 'Y'       REDSPY
C*
C*****
C* OTHERWISE ENTER WAS PRESSED
C*****
C*
C                      OTHER
C* Check the input fields for errors
C                      EXSR CHKERR
C*
C* If error exists redisplay the panel with error fields
C* in reversed image
C          PNLERR     IFEQ 'Y'
C                      MOVE 'Y'       REDSPY
C                      END
C*
C* End the select statement and the do while loop
C                      ENDSL
C                      END
C*
C* Read subfile and get first selected record, set the
C* parameters from selected record for call to API
C                      EXSR GETSEL
C*
C* Set remaining parameters and call the
C* control calendar API
C                      EXSR CTLCAL
C*
C                      SETON           LR
C                      RETRN
C*

```

Figure 2-3 (Part 6 of 17). RPG Program Called by SLTQRYITM CL Program

```

C*****
C* END OF MAIN LINE CODE
C*****
C/EJECT
C*****
C*          A D D R E C
C* SUBROUTINE TO ADD A RECORD TO THE SUBFILE
C*****
C          ADDREC  BEGSR
C*
C* Increment the subfile record number
C          ADD 1      SUB#
C*
C* Check if record has already been marked as an error
C          Z-ADD1      I      20
C          REC#      LOKUPRCD,I      32
C          *IN32      IFEQ '1'
C*
C* If more than one valid option reverse image option field
C          ONEOPT      IFEQ 'N'
C          OPT,I      ANDEQ'1'
C          MOVE '0'      *IN84
C*
C          ELSE
C          MOVE '1'      *IN84
C          LNNBR      IFEQ 0
C          SUB#      ADD #OPTLN      LNNBR
C          Z-ADD#OPTPS      PSNBR
C          END
C          END
C*
C* Load option field from error option array
C          MOVE OPT,I      POPT
C*
C* Load option field with a blank
C          ELSE
C          MOVE ' '      POPT
C          MOVE '0'      *IN84
C          END
C          MOVE '1'      *IN97
C*
C* Determine type of item
C          Z-ADD1      I
C          QCITYP      LOKUPITM,I      32
C          *IN32      IFEQ '1'
C          MOVE ITS,I      PITEM
C          END
C*
C* Convert current date to MM/DD/YY format
C          MOVE QCDATE      MMDD      4
C          MOVE LQCDATE      YY
C          MOVE LMMDD      MM
C          MOVE MMDD      DD
C          MOVE STRDAT      PSDATE
C*

```

Figure 2-3 (Part 7 of 17). RPG Program Called by SLTQRYITM CL Program

```

C* Convert start time to HH:MM in 12hr format
C          QCITYP      IFNE ITM,3
C          MOVE LQCSTIM      PSTIME
C*
C          ZHOURS      IFLE 12
C          MOVE #AM          PSTIME
C          ELSE
C          ZHOURS      SUB 12      ZHOURS
C          MOVE #PM          PSTIME
C          END
C*
C          ZHOURS      IFLT 10
C          MOVE L' '          PHOURS
C          END
C*
C          ELSE
C          MOVE *BLANK      PSTIME
C          END
C*
C* Set the calendar name, text and record number
C          MOVE QCCAL      PCALNM
C          MOVE QCODEN      PCALDN
C          MOVE QCODGN      PCALDG
C          MOVE LQCTXT1      PTEXT
C          Z-ADDREC#      PRECNO
C*
C* Write record to subfile
C          WRITESUB
C          MOVE '0'          *IN97
C*
C          ENDSR
C/EJECT
C*****
C*          B L D S U B
C* SUBROUTINE TO BUILD THE SUBFILE PORTION OF THE PANEL
C*****
C          BLDSUB      BEGSR
C*
C* Position to file record number
C          REC#          SETLLQRYITMF          30
C          *IN30          IFEQ '1'
C          Z-ADD0          SUB#
C*
C* Clear the record subfile
C          MOVE '1'          *IN99
C          WRITESUBCTL
C          MOVE '0'          *IN99
C*
C* Loop until no more records in QRYITMF file
C          MOVE 'Y'          LOOP      1
C          LOOP          DOWEQ'Y'
C*
C          READ QRYITMF          30
C*

```

Figure 2-3 (Part 8 of 17). RPG Program Called by SLTQRYITM CL Program


```

C* Check if page is full or end of file
C      *IN30      IFEQ '1'
C      SUB#       OREQ #PAGE
C              MOVE 'N'          LOOP
C*
C* Write the record to the subfile and increment the
C* relative record number
C              ELSE
C              EXSR ADDREC
C              ADD 1          REC#
C              END
C              END
C              END
C*
C* Check if there are more records in the file
C      Z-ADDREC#      SAVREC 40
C      READ QRYITMF          30
C*
C      *IN30      IFEQ '1'
C              MOVE 'Bottom ' LSTMSG
C              ELSE
C              MOVE 'More...' LSTMSG
C              END
C              Z-ADDSAVREC      REC#
C*
C              ENDSR
C/EJECT
C*****
C*              C H K E R R
C* SUBROUTINE TO CHECK INPUT FIELDS FOR ERRORS
C*****
C      CHKERR      BEGSR
C*
C* Signal not to redisplay the panel
C              MOVE 'N'          PNLERR
C*
C* Reset reverse image indicators
C              MOVE '0'          *IN81
C              MOVE '0'          *IN82
C              MOVE '0'          *IN83
C              MOVE '0'          *IN84
C*
C* Reset cursor location
C              Z-ADD0          LNNBR
C              Z-ADD0          PSNBR
C*
C* Remove the messages from this programs message queue
C              EXSR RMVMSG
C*

```

Figure 2-3 (Part 9 of 17). RPG Program Called by SLTQRYITM CL Program

```

C* Validate calendar view type
C          Z-ADD1          I
C          PVTYP          LOKUPVTP,I          32
C          *IN32          IFEQ '0'
C          MOVE '1'          *IN81
C          Z-ADD#VTPLN          LNNBR
C          Z-ADD#VTPPS          PSNBR
C          MOVE 'Y'          PNLERR
C*
C* Send the message to this programs message queue
C          MOVELERR,1          ERRMSG 50
C          EXSR SNDMSG
C          END
C*
C* Validate number of columns
C*
C          PCOLS          IFEQ *BLANK
C          Z-ADD0          ZCOLS
C*
C          ELSE
C          MOVELPCOLS          CCOLS
C          Z-ADD1          I
C          CCOLS          LOKUPNBR,I          32
C*
C          *IN32          IFEQ '0'
C          ZCOLS          ORLT #MNCOL
C          ZCOLS          ORGT #MXCOL
C          MOVE '1'          *IN82
C*
C          PNLERR          IFNE 'Y'
C          Z-ADD#CLSLN          LNNBR
C          Z-ADD#CLSPPS          PSNBR
C          MOVE 'Y'          PNLERR
C          END
C*
C* Send the message to this programs message queue
C          MOVELERR,2          ERRMSG
C          EXSR SNDMSG
C          END
C          END
C*
C* Validate the time interval
C          PINTVL          IFEQ *BLANK
C          Z-ADD0          ZINTVL
C*
C          ELSE
C          MOVELPINTVL          PINTV1 1
C          MOVE PINTVL          PINTV2 1
C*

```

Figure 2-3 (Part 10 of 17). RPG Program Called by SLTQRYITM CL Program

```

C* Change blanks to zeros
C          SELEC
C          PINTV1  WHEQ *BLANK
C          MOVE '0'      CINTV1
C          MOVE PINTV2   CINTV2
C          PINTV2  WHEQ *BLANK
C          MOVE '0'      CINTV1
C          MOVE PINTV1   CINTV2
C          OTHER
C          MOVE PINTVL   CINTVL
C          ENDSL
C*
C          Z-ADD1      I
C          Z-ADD1      II
C          CINTV1     LOKUPNBR,I      31
C          CINTV2     LOKUPNBR,II     32
C*
C          *IN31      IFEQ '0'
C          *IN32      OREQ '0'
C          ZINTVL     ORLT #MNINT
C          ZINTVL     ORGT #MXINT
C          MOVE '1'      *IN83
C*
C          PNLERR     IFNE 'Y'
C          Z-ADD#INTLN  LNNBR
C          Z-ADD#INTPS  PSNBR
C          MOVE 'Y'      PNLERR
C          END
C*
C* Send the message to this programs message queue
C          MOVE LERR,3  ERRMSG
C          EXSR SNDMSG
C          END
C          END
C*
C* Validate the option field in the list
C          EXSR VLDLST
C*
C* If cursor is not positioned yet (no errors), then set it
C          PNLERR     IFEQ 'N'
C          CLOC       DIV 256      LNNBR
C          MVR        PSNBR
C          LNNBR     IFGT #OPTLN
C          Z-ADD#OPTLN  LNNBR
C          Z-ADD#OPTPS  PSNBR
C          END
C          END
C*
C          ENDSR
C/EJECT

```

Figure 2-3 (Part 11 of 17). RPG Program Called by SLTQRYITM CL Program

```

C*****
C*          C T L C A L
C* SUBROUTINE TO CALL THE CONTROL CALENDAR API (QOCTLCL)
C*****
C          CTLCAL    BEGSR
C*
C* Set the view type, columns, and interval parameters
C          PVTTYPE   IFEQ *BLANK
C                      MOVELNONE10    VWTYPE 10
C                      ELSE
C                      MOVELPVTTYPE    VWTYPE
C                      END
C                      Z-ADDZCOLS      COLMNS
C                      Z-ADDZINTVL     INTRVL
C*
C* Set remaining parameters not used in this program
C                      MOVE *BLANKS    GRPCAL 26
C                      MOVE *BLANKS    DISTL 16
C                      Z-ADD0          ERRCD
C*
C* Call the control calendar API
C          CALL 'QOCTLCL'
C          PARM          VWTYPE
C          PARM          CURDAT
C          PARM          STRTIM
C          PARM          COLMNS
C          PARM          INTRVL
C          PARM          CALNDR
C          PARM          CALNBR
C          PARM          GRPCAL
C          PARM          DISTL
C          PARM          ERRCD
C*
C          ENDSR
C/EJECT
C*****
C*          G E T S E L
C* SUBROUTINE TO GET THE SELECTED RECORD
C*****
C          GETSEL     BEGSR
C*
C* If record not selected
C          OPT,1      IFNE '1'
C                      Z-ADD0          CALNBR
C                      MOVELNONE10    CURDAT
C                      MOVELNONE10    STRTIM
C*
C* Record was selected
C          ELSE
C          REC#       Z-ADDRCD,1      REC#
C          REC#       CHAINQRYITMF    30
C          Z-ADD1     CALNBR
C*

```

Figure 2-3 (Part 12 of 17). RPG Program Called by SLTQRYITM CL Program

```

C* Build a calendar name
C          MOVELQCCAL      CALNAM
C          MOVELQCODEN     CALDEN
C          MOVELQCODGN     CALDGN
C*
C* Convert current date to CYYMMDD
C          MOVE'0'        CENTRY
C          MOVE QCDATE     YYMMDD
C*
C* Convert start time to HHMMSS (when item is not a reminder)
C          PITEM          IFNE ITS,3
C          MOVELQCSTIM     HOURS
C          MOVE QCSTIM     MINS
C          MOVE '00'       SECS
C          ELSE
C          MOVELNONE10     STRTIM
C          END
C          END
C*
C          ENDSR
C/EJECT
C*****
C*          R E F R S H
C* SUBROUTINE TO REFRESH THE PANELS
C*****
C          REFRSH      BEGSR
C*
C* Refresh the panel variables
C          MOVE *BLANK  PVTTYPE
C          MOVE *BLANK  PCOLS
C          MOVE *BLANK  PINTVL
C*
C* Refresh reverse image and cursor position indicators
C          MOVE '0'     *IN81
C          MOVE '0'     *IN82
C          MOVE '0'     *IN83
C*
C* Refresh the option fields by rebuilding the record subfile
C          MOVE 'N'     ONEOPT
C          REC#         IFGT #PAGE
C          SUB #PAGE    REC#
C          ELSE
C          Z-ADD1       REC#
C          END
C          MOVE *ZERO   RCD
C          MOVE *BLANK  OPT
C          EXSR BLDSUB
C*
C* Remove the messages from this programs message queue
C          EXSR RMVMSG
C*

```

Figure 2-3 (Part 13 of 17). RPG Program Called by SLTQRYITM CL Program

```

C* Reset cursor position
C           Z-ADD0           LNNBR
C           Z-ADD0           PSNBR
C*
C           ENDSR
C/EJECT
C*****
C*           R M V M S G
C* SUBROUTINE TO REMOVE MESSAGES FROM THIS PROGRAMS MESSAGE QUEUE
C*****
C           RMVMSG   BEGSR
C*
C* Call SENDMSG to run the RMVMSG command
C           CALL 'SENDMSG'
C           PARM 'Y'           DLTMSG
C           PARM *BLANK       MSGID
C           PARM *BLANK       MSGF
C           PARM *BLANK       MSGL
C           PARM *BLANK       ERRMSG
C           PARM *BLANK       MSGTYP
C*
C           ENDSR
C/EJECT
C*****
C*           S N D M S G
C* SUBROUTINE TO SEND A MESSAGE TO THIS PROGRAMS MESSAGE QUEUE
C*****
C           SNDMSG   BEGSR
C*
C* Call SENDMSG to run the SNDPGMSG command
C           CALL 'SENDMSG'
C           PARM 'N'           DLTMSG 1
C           PARM *BLANK       MSGID 7
C           PARM *BLANK       MSGF 10
C           PARM *BLANK       MSGL 10
C           PARM               ERRMSG
C           PARM '*DIAG'      'MSGTYP 8
C*
C           ENDSR
C/EJECT
C*****
C*           V L D L S T
C* SUBROUTINE TO VALIDATE THE OPTION FIELD IN THE LIST
C*****
C           VLDLST   BEGSR
C*
C* Validate the option field
C           MOVE 'Y'           LOOP
C           LOOP           DOWEQ'Y'
C*

```

Figure 2-3 (Part 14 of 17). RPG Program Called by SLTQRYITM CL Program

```

C* Check if any subfile records changed
C          READCSUB                                31
C*
C          *IN31   IFEQ '1'
C          MOVE 'N'          LOOP
C*
C* If option is blank then remove record number from
C* the error record array
C          ELSE
C          POPT      IFEQ ' '
C          Z-ADD1          I
C          PRECNO    LOKUPRCD,I                                32
C          *IN32    IFEQ '1'
C          Z-ADDI          II      20
C*
C* Reorganize the arrays
C          RCD,I     DOWNE*ZERO
C          ADD 1          II
C          Z-ADDRCD,II  RCD,I
C          MOVE OPT,II  OPT,I
C          ADD 1          I
C          END
C          END
C*
C* Else option is a one or not valid, so add the record
C* to the error record array
C          ELSE
C          Z-ADD1          I
C          PRECNO    LOKUPRCD,I                                32
C*
C          *IN32    IFEQ '1'
C          Z-ADDPRECNO  RCD,I
C          MOVE POPT   OPT,I
C*
C          ELSE
C          *ZERO      LOKUPRCD,I                                32
C          Z-ADDPRECNO  RCD,I
C          MOVE POPT   OPT,I
C          END
C          END
C          END
C          END
C*
C* Send the messages to this programs message queue
C          Z-ADD1          I
C          RCD,I     DOWNE*ZERO
C          OPT,I     IFNE '1'
C          MOVE LERR,4  ERRMSG
C          EXSR SNDMSG
C          MOVE 'Y'     PNLERR
C          END
C          ADD 1          I
C          END
C*

```

Figure 2-3 (Part 15 of 17). RPG Program Called by SLTQRYITM CL Program

```

C* If a panel error exists then option '1's are not to
C* reversed imaged
C          PNLERR      IFEQ 'Y'
C          MOVE 'N'      ONEOPT
C          REC#        IFGT #PAGE
C          SUB #PAGE      REC#
C          ELSE
C          Z-ADD1        REC#
C          END
C          EXSR BLDSUB
C          END
C*
C* If no panel error check if more then one selection made
C          *IN07      IFEQ '0'
C          *IN08      ANDEQ'0'
C          PNLERR      IFEQ 'N'
C          OPT,2      ANDEQ'1'
C          MOVE LERR,5      ERRMSG
C          EXSR SNDMSG
C          MOVE 'Y'      PNLERR
C          MOVE 'Y'      ONEOPT
C          Z-ADDRCD,1      REC#
C          EXSR BLDSUB
C          END
C          END
C*
C          ENDSR
C/EJECT

```

```

** ERR - ERROR MESSAGES
View type is not valid.
Number of columns is not valid.
Time interval is not valid.
Option selected is not valid.
More then one option selected.

```

```

** VTP - VIEW TYPES
*WEEKLY
*DAILY
*REMINDER
*SIXMONTH

```

```

** ITM - ITEM TYPES
*MEETING
*EVENT
*REMINDER
*JOB
*S36PRC

```

Figure 2-3 (Part 16 of 17). RPG Program Called by SLTQRYITM CL Program


```

** ITS - ITEM TYPES (Abbreviated)
MTG
EVT
RMD
JOB
PRC
** NBR - NUMBERS
0
1
2
3
4
5
6
7
8
9

```

Figure 2-3 (Part 17 of 17). RPG Program Called by SLTQRYITM CL Program

The following example shows the DDS source used by the RPG program SLTQRYIT.

```

A*****      START OF SPECIFICATIONS      *****/
A*                                                    */
A* DEVICE FILE NAME: DQRYITM                    */
A*                                                    */
A* DEVICE FILE TYPE: DISPLAY FILE              */
A*                                                    */
A* FUNCTION: Select query calendar item display file */
A*                                                    */
A* RECORD FORMAT NAMES:                        */
A*     RECORD1 Select queried calendar items    */
A*     RECORD2 Function keys                   */
A*     RECORD3 Function keys (no items to display) */
A*     SUB      Subfile for select queried item calendar */
A*     SUBCTL   Control record for select queried item calendar */
A*     MSG      Subfile for messages           */
A*     MSGCTL   Control record for message subfile */
A*                                                    */
A*****/
A*****/
A* File level keywords                            */
A*****/
A                DSPSIZ(24 80 *DS3)
A                INDARA
A                CA03(03 'Exit')
A                CA05(05 'Refresh')
A*                                                    */

```

Figure 2-4 (Part 1 of 4). DDS Source for RPG Program SLTQRYIT

```

A*****
A* Select Queried Calendar Items Panel */
A*****
A          R PANEL1
A
A          ROLLUP(07 'Rollup')
A          ROLLDOWN(08 'Rolldown')
A          CSRLOC(LNNBR PSNBR)
A          OVERLAY
A          BLINK
A          LNNBR          3S 0H
A          PSNBR          3S 0H
A
A          4 2'Type choices, press Enter.'
A          5 4'View Type . . . . . '
A          7 4'Number of columns . . . . . '
A          8 4'Time interval . . . . . '
A          PVTTYPE          10A B 5 37
A 81          DSPATR(RI)
A N81         DSPATR(UL)
A          10 2'Type options, press Enter.'
A          11 4'1=Select'
A          13 2'Opt'
A          DSPATR(HI)
A          DSPATR(UL)
A          PCOLS          1A B 7 37
A 82          DSPATR(RI)
A N82         DSPATR(UL)
A          13 7'Itm'
A          DSPATR(UL)
A          DSPATR(HI)
A          13 12'S-Date '
A          DSPATR(UL)
A          DSPATR(HI)
A          13 22'S-Time'
A          DSPATR(UL)
A          DSPATR(HI)
A          13 30'Calendar '
A          DSPATR(UL)
A          DSPATR(HI)
A          13 41'User Id '
A          DSPATR(UL)
A          DSPATR(HI)
A          13 50'Address '
A          DSPATR(UL)
A          DSPATR(HI)
A          12 41'-----Owner-----'
A          DSPATR(HI)
A          13 60'Text
A          DSPATR(UL)
A          DSPATR(HI)
A          2 24'Display Queried Item Calendar'
A          DSPATR(HI)

```

Figure 2-4 (Part 2 of 4). DDS Source for RPG Program SLTQRYIT

```

A          PINTVL          2A B 8 37
A 83          DSPATR(RI)
A N83         DSPATR(UL)
A           5 49' *DAILY, *WEEKLY, *REMINDER, '
A           6 49' or *SIXMONTH'
A           7 49'5, 6, or 7'
A           8 49'5-60'
A*
A*****
A* Function keys (with items to display)
A*****
A          R PANEL2
A
A              OVERLAY
A              BLINK
A              22 2'F3=Exit  F5=Refresh
A
A
A              23 2'
A
A
A          LSTMSG          8A 0 21 70DSPATR(HI)
A*
A*****
A* Function keys (no items to display)
A*****
A          R PANEL3
A
A              OVERLAY
A              BLINK
A              22 2'F3=Exit  F5=Refresh
A
A
A              23 2'
A
A
A              14 4'(No items to display)
A              21 70'Bottom
A              DSPATR(HI)
A*
A*****
A* List of items to select (subfile)
A*****
A          R SUB          SFL
A          POPT          1A B 14 3
A 97          DSPATR(UL)
A 84          DSPATR(RI)
A          PITEM          3A 0 14 7
A          PSDATE          8A 0 14 12
A          PSTIME          6A 0 14 22
A          PCALNM          10A 0 14 30
A          PCALDN          8A 0 14 41
A          PCALDG          8A 0 14 50
A          PTEXT          20 0 14 60
A          PRECNO          4D 0H
A*

```

Figure 2-4 (Part 3 of 4). DDS Source for RPG Program SLTQRYIT

```

A*****/
A* List of items to select (control record for subfile) */
A*****/
A          R SUBCTL                SFLCTL(SUB)
A                                OVERLAY
A 96                                SFLDSP
A 96                                SFLDSPCTL
A 99                                SFLINZ
A 99                                SFLCLR
A                                SFLRNA
A                                SFLSIZ(0007)
A                                SFLPAG(0007)
A*                                     */
A*****/
A* Message subfile */
A*****/
A          R MSG                    SFL
A                                SFLMSGRCD(24)
A          MSGK                    SFLMSGKEY
A          PGMQ                    SFLPGMQ
A*                                     */
A*****/
A* Message subfile (control record for subfile) */
A*****/
A          R MSGCTL                SFLCTL(MSG)
A                                LOCK
A                                OVERLAY
A                                SFLDSP
A                                SFLDSPCTL
A                                SFLINZ
A 94                                SFLEND
A                                SFLSIZ(0010)
A                                SFLPAG(0001)
A          PGMQ                    SFLPGMQ
A*                                     */

```

Figure 2-4 (Part 4 of 4). DDS Source for RPG Program SLTQRYIT

Figure 2-5 shows the CL program SENDMSG called by the RPG program SLTQRYIT.

```

/**** START OF SPECIFICATIONS *****/
/*
/* PROCESSOR: CL PROGRAM
/*
/* MODULE NAME: SENDMSG
/*
/* DESCRIPTIVE NAME: Sends or removes messages to/from the previous
/* programs message queue.
/*
/* FUNCTION/PURPOSE:
/* To stack messages in the previous programs message queue or
/* remove all the messages in the previous programs message queue.
/*
/* DEPENDENCIES:
/* SNDPGMMSG CL command.
/* RMVMSG CL command.
/*
/* INPUT:
/* RMVMSG - The 1-byte variable ('Y' or 'N')
/* MSGID - The 7-byte message identifier
/* MSGF - The 10-byte message file for the message id
/* MSGL - The 10-byte library name where the message file can
/* be found
/* ERRMSG - The 50-byte message used instead of MSGID
/* MSGTYPE - The 8-byte message type
/*
/*****/
PGM PARM(&RMVMSG &MSGID &MSGF &MSGL &ERRMSG +
&MSGTYPE)

/*
/* Declare parameters and local variables
/*
DCL VAR(&RMVMSG) TYPE(*CHAR) LEN(1)
DCL VAR(&MSGID) TYPE(*CHAR) LEN(7)
DCL VAR(&MSGF) TYPE(*CHAR) LEN(10)
DCL VAR(&MSGL) TYPE(*CHAR) LEN(10)
DCL VAR(&ERRMSG) TYPE(*CHAR) LEN(50)
DCL VAR(&MSGTYPE) TYPE(*CHAR) LEN(8)
DCL VAR(&BLANK) TYPE(*CHAR) LEN(50)

/*
/* Check if the all messages are to be deleted from previous
/* programs message queue
/*
IF COND(&RMVMSG *EQ 'Y') THEN(DO)
RMVMSG PGMQ(*PRV) CLEAR(*ALL)
MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(RMVERR))
GOTO CMDLBL(END)
ENDDO

```

Figure 2-5 (Part 1 of 2). CL Program SENDMSG Called by SLTQRYIT RPG Program

```

/*                                                                    */
/* Check if message id or 50 character message is to be sent to      */
/* the previous programs message queue                                */
/*                                                                    */
      IF          COND(&ERRMSG *NE &BLANK) THEN(DO)
      SNDPGMMSG  MSG(&ERRMSG) TOPGMQ(*PRV) MSGTYPE(&MSGTYPE)
      MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(SNDERR))
      GOTO       CMDLBL(END)
      ENDDO

      ELSE       CMD(DO)
      SNDPGMMSG  MSGID(&MSGID) MSGF(&MSGL/&MSGF) +
                MSGTYPE(&MSGTYPE)
      MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(SNDERR))
      GOTO       CMDLBL(END)
      ENDDO

/*                                                                    */
/* Signal error sent by RMVMSG and quit                                */
/*                                                                    */
RMVERR:    SNDPGMMSG  MSG('Fatal error on RMVMSG command. See +
                    previous messages.')
          GOTO       CMDLBL(END)

/*                                                                    */
/* Signal error sent by SNDPGMMSG and quit                            */
/*                                                                    */
SNDERR:    SNDPGMMSG  MSG('Fatal error on SNDPGMMSG command. See +
                    previous messages.')
          GOTO       CMDLBL(END)

/*                                                                    */
/* Signal necessary errors and quit                                    */
/*                                                                    */
END:       ENDPGM

```

Figure 2-5 (Part 2 of 2). CL Program SENDMSG Called by SLTQRYIT RPG Program

Chapter 3. Directory Services

This chapter describes the directory services available to AS/400 system users. The following topics are discussed:

- System directory objects
- Naming conventions
- Service concepts
- Security considerations
- Control language (CL) commands used with other program applications to maintain the system directory

Directory Services Overview

Directory services provides support to maintain the system distribution directory. Directory information is used by:

- OS/400 SNADS component
- OS/400 distribution services
- OS/400 object distribution
- OfficeVision/400
- OS/400 library service

The system distribution directory identifies users as local, remote, or indirect. A **local user** is a user who signs on to your system. A **remote user** is a user who signs on to another system in a network to which your system belongs. An **indirect user** is a local user who receives electronic mail in printed form rather than signing on to the system to receive mail. The directory contains additional information about the users of your system, such as mailing address and telephone number.

Directory services also maintains distribution lists. A **distribution list** is a collection of system distribution directory entries used to send information to a group of users instead of to one user at a time. A distribution list may contain the following:

- Local users
- Remote users
- Remote distribution lists

Directory services also provides nickname support. Nicknames provide the user with a short form for addressing another user or a distribution list. You can use a nickname with OfficeVision/400, or with the Work with Directory (WRKDIR) and Work with Distribution List (WRKDSTL) commands. Nicknames can only be specified from panels that support nicknames.

The directory services also provides support for department information and for members of departments. You can use the Work with Directory (WRKDIR) and Work with Distribution List (WRKDSTL) commands to access the department functions.

Directory services provides support to enter an Open Systems Interconnection (OSI) X.400 originator/recipient name (O/R name) using the directory CL commands (ADDDIRE, CHGDIRE, DSPDIR, and WRKDIR).

Directory services provides support for directory shadowing. **Shadowing** is the ability to supply the directory data to collecting systems. Whenever changes occur to the data, these changes can be supplied to the collecting systems within the network. The systems that you request directory data from are **suppliers**, and the systems that receive directory data from your system are **collectors**. To receive directory data from another system, the collecting system must request the data. The system administrator identifies which systems they want to shadow, when to shadow the systems, and at what frequency these systems need to be shadowed.

For more information about shadowing, refer to the *Distribution Services Network Guide*.

System Distribution Directory

The system distribution directory contains the user ID, user address, and description for users authorized to send and receive distributions and other information about users. The **user ID** and **address** are used together in distributions to identify a specific a user and must be unique within the network. The user ID and address can be changed using the Rename Directory Entry (RNMDIRE) command or option 7 (Rename) on the Work with Directory (WRKDIR) display. See the *CL Reference* manual for details on these commands.

Every user ID and address must have at least one description, but can have more than one. The description is the text for each entry in the directory that describes the user. The description does not have to be unique across users. If the user ID and address has more than one description, then each description for that user must be unique.

A local directory entry must have a user profile defined with the entry. A user profile must exist on the system before adding a local user to the directory. A user profile can be associated with only one user ID and address.

A user must have an entry in the system distribution directory to use OfficeVision/400 and object distribution. When enrolling new users on the AS/400 system, it may be desirable to limit the user profile name to eight characters or less, and to make the user ID the same as the user profile for simplicity. However, there is no requirement that they match.

Distribution Lists

Distribution lists are used by office users and object distribution users to make it easier to distribute items to the same set of users more than once with a single name. When distributing an item, a user can specify entries for one or more distribution lists, one or more users, or any combination.

Local user IDs must already be defined in the system distribution directory before being added to a distribution list. Remote user IDs or remote distribution list IDs must be defined in the system distribution directory, or an *ANY or *ANY *ANY entry for that remote node must be defined. (*ANY is the default entry for a user ID of a remote user or distribution list. *ANY *ANY is the default entry for both the user ID and address of a remote user or distribution list.) If shadowing is used, remote users do not need to be defined in the directory.

Every distribution list must have a description. The creator of a distribution list becomes the owner of that list. Only the owner, administrator (*SECADM special

authority), or a security officer can change or delete a distribution list. Anyone can view a distribution list. There is no limit to either the number of distribution lists a person can own or the number of entries in a specific list. The ownership of a distribution list cannot be changed to another user. When a user is removed from the directory and that user owns distribution lists, the distribution list owner is automatically changed to QDFTOWN.

When a directory entry is removed, the entries that exist in distribution lists are deleted. When an *ANY entry is removed and an *ANY *ANY entry exists, the entries that exist in distribution lists because of the *ANY entry are not deleted. The description in the distribution list is changed to the description of the *ANY *ANY entry.

Directory Services Objects

This section discusses directory files, nickname files, recipient indexes, distribution list files, and object naming conventions.

All directory services files are defined and shipped with the system in library QUSRSYS. Directory services files use the following naming conventions:

QAOKP*nnv* Directory physical files
QAOKL*nnv* Directory logical files
QAOKS*nnv* Directory search files

where *nn* is a system-assigned number and *v* is a version identifier. These files should not be deleted. Refer to Chapter 9, "Data Backup, Recovery, and Storage Management," for journaling information.

The naming convention for model output files in library QSYS is QAOSxxxx.

Directory Files

The directory is contained in physical database files connected by several logical files. The following database files are stored in library QUSRSYS:

QAOKP01A Local Document Interchange Architecture (DIA) definition file
QAOKP04A Directory routing file name
QAOKP05A Directory department content file
QAOKP06A Nickname file
QAOKP08A Directory entry description
QAOKP09A Directory fields
QAOKPX4A X.400 originator/recipient name
QAOKPCLA Directory shadow collectors file
QAOKPLCA Directory location file
QAOKPLGA Directory shadow change log file
QAOKPSPA Directory shadow suppliers file

Note: Customer programs must not access the directory files directly, because it is an IBM-reserved right to change the file format, the method, or location of storing directory data.

Output File

The Display Directory (DSPDIR) command can be used to create a user-named database file that contains the information from the system distribution directory. The output file does not change, but fields can be added.

You can specify three types of output files using the OUTFILFMT parameter. The default is OUTFILFMT(*TYPE1). The other types are OUTFILFMT(*TYPE2) and OUTFILFMT(*TYPE3).

***TYPE1:** When you specify OUTFILFMT(*TYPE1), the output file contains either basic or full information about the directory entries, depending on the user ID and address (USRID), user profile (USER), and DETAIL parameters specified on the DSPDIR command. Use this output file when you write a program that needs directory information.

If the USRID or USER parameters specified a unique entry, one database record is written with the full directory details for that entry. If more than one description exists for the specified USRID or USER, the output file contains a detail record for each description.

The model output file used is QAOSDIRO in library QSYS.

When the USRID is specified as, or takes the default value of *ALL, the DETAIL parameter determines what data is written to the database file. If DETAIL(*BASIC) was specified, the database records only contain the user ID, address, and description. If DETAIL(*FULL) was specified, a full detail record is built for all users. When multiple descriptions exist in the directory for a given user ID and address, one record is written for each description. The only data that is different between these records is the description field. Refer to the "Display Directory (DSPDIR) Command" on page 3-13 for more information. Some of the field lengths in this file are not the full field lengths supported with *TYPE2.

***TYPE2:** When you specify OUTFILFMT(*TYPE2), the data is put out to the output file in a different format because of field length changes and additional fields.

If you specify DETAIL(*BASIC), the model output file used is QAOSDIRB in library QSYS. The records contain the user ID, address, and description.

If you specify DETAIL(*FULL) or a specific user, the model output file used is QAOSDIRF in library QSYS. A full detail record is built with all the additional fields included, except X.400 fields.

***TYPE3:** When you specify OUTFILFMT(*TYPE3), the output file contains either basic or full information about file formats.

If you specify DETAIL(*BASIC), the model output file used is QAOSDIRB in library QSYS. The records contain the user ID, address, and description.

If you specify DETAIL(*FULL), an output file of model format QAOSDIRX in library QSYS is generated. The records contain the directory fields plus the X.400 originator/recipient name (O/R name) fields.

(See Appendix C, "Office Services Output File Considerations," for the data description specifications of the output file.)

Distribution Recipient Index

When a local user is enrolled in the system distribution directory, a distribution recipient index (DRX) is created in the QUSRSYS library. This index is used by document distribution services and object distribution to control object distribution and document distribution.

(See “Distribution Recipient Index” on page 4-6 for more information.)

Distribution List Files

Two physical files contain distribution lists. Several logical files are used to access the information contained in the physical files. These files are stored in library QUSRSYS.

QAOKP02A Distribution list definition file

QAOKP03A Distribution list content file

Note: Customer programs must not access the distribution list files directly because it is an IBM-reserved right to change the format of the files or the method or location of storing distribution list data.

Output File

The Display Distribution List (DSPDSTL) command can be used to create a user-named database file that contains the information about distribution lists. If the OUTPUT (*OUTFILE) parameter is specified, a user-named database file is required for the *OUTFILE parameter. If the file does not exist, a new file is created with public authority *EXCLUDE.

The record formats are defined in a physical file QAOSDSTO in library QSYS. A record is written each time an entry occurs in a distribution list. This output file may be used when user application programs need distribution list information.

The output file contains a header record for each distribution list being written. This header record contains the list ID and description for the output list. The header record may be followed by detail records that contain the user ID/address and description for each of the entries contained in the distribution list.

The database file contents are determined by the LSTID and DETAIL parameters on the DSPDSTL command. If the LSTID parameter specifies a list ID, the entries within that list are written to the database file. If the LSTID parameter specifies *ALL and the DETAIL parameter specifies *BASIC, each list ID and description within the directory is written to the database file (a header record is written for each distribution list contained in the directory). When LSTID(*ALL) is specified and DETAIL(*FULL) is specified, a header record is written for the first distribution list contained in the directory, followed by detail records for all entries in the list. A header record is then written for the next distribution list, again followed by detail records for all entries in that list. This continues until all distribution lists have been written to the output file. Refer to the “Display Distribution List (DSPDSTL) Command” on page 3-23 for more information.

(See Appendix C, “Office Services Output File Considerations,” for the data description specifications of the output file.)

The naming convention for output files is the responsibility of the user.

Directory Services Commands

An AS/400 system provides the following support to work with the system distribution directory and distribution lists:

- Interactive commands to make use of displays that help you manage the directory and distribution lists. These commands must be used at a work station.
- Other commands that allow you to manage the directory and associated distribution lists without a display interface or in a batch job. These commands could be used to:
 - Run in an input stream from a batch job queue
 - Combine with other commands and compile into a CL program that together with RPG, COBOL, or other programs can create a batch application to manage the directory or distribution lists
 - Create a customer application to provide an installation tailored interactive menu/display driven interface

The following commands are used in the maintenance of the system directory:

| | |
|------------------|-------------------------------|
| ADDIRE | Add Directory Entry |
| CHGDIRA | Change Directory Attributes |
| CHGDIRE | Change Directory Entry |
| RMVDIRE | Remove Directory Entry |
| RNMDIRE | Rename Directory Entry |
| WRKDIR | Work with Directory |
| DSPDIR | Display Directory |
| WRKDIRLOC | Work with Directory Locations |

Note: See the *Distribution Services Network Guide* for information about directory locations.

The following commands are used for maintenance of distribution lists:

| | |
|-----------------|--------------------------------|
| CRTDSTL | Create Distribution List |
| DLTDSTL | Delete Distribution List |
| ADDSTLE | Add Distribution List Entry |
| RMVDSTLE | Remove Distribution List Entry |
| DSPDSTL | Display Distribution List |
| WRKDSTL | Work with Distribution Lists |

The following commands are used for directory shadowing. See the *Distribution Services Network Guide* for information on these commands.

| | |
|------------------|------------------------------------|
| ADDIRSHD | Add Directory Shadow System |
| CHGDIRSHD | Change Directory Shadow System |
| CPYFRMDIR | Copy from Directory |
| CPYTODIR | Copy to Directory |
| ENDDIRSHD | End Directory Shadowing |
| RMVDIRSHD | Remove Directory Shadow System |
| STRDIRSHD | Start Directory Shadowing |
| WRKDIRSHD | Work with Directory Shadow Systems |

The WRKDIR, WRKDSTL, WRKDIRLOC, and WRKDIRSHD commands can only be used interactively. The DSPDIR and DSPDSTL commands can be used inter-

actively, but can also be used in batch application programs when the print or the output file parameter is specified. The ADDDIRE, CHGDIRE, RMVDIRE, RNMDIRE, CRTDSTL, DLTSTL, ADDDSTLE, and RMVDSTLE commands are normally used in batch application programs to update the system directory or distribution lists, but can also be used interactively.

An application program using these commands can be used to update system directories and distribution lists at remote locations. The DSPDIR and DSPDSTL commands provide support to write directory entries and distribution lists to output files. These output files can then be used by the application programs to update directories and distribution lists in a central directory. The application programs can be submitted as batch jobs and sent to a remote location as an input stream using the object distribution functions of an AS/400 system to update remote directories.

In some applications, much of the data needed in the directory already exists in customer files, such as personnel databases. The CL commands can be used along with other high-level language programs to provide a batch application to extract the customer data and build the directory. Thus, an operator would not have to type the data again.

Other Directory Services Topics

Information discussed includes security considerations, including restrictions that apply to users and security administrators and related directory entries.

Security Considerations

This section discusses directory entries for folders and documents, user restrictions, security administrator restrictions, and IBM profile directory entries.

Required Directory Entries

All of the following applications require a local directory entry in the system directory.

Folder Ownership and Uses

To own folders, a directory entry is required. The personal computer uses folders as PC directories.

Document and PC File Ownership and Uses

To own documents, a directory entry is required. The personal computer uses documents as PC files.

OfficeVision/400 Uses

A local directory entry is required for a user to be enrolled in OfficeVision/400. When the system administrator enrolls a user in OfficeVision/400, a directory entry and user profile can be created automatically.

Object Distribution Commands

Object distribution provides a method for sending objects and messages from one user to another user or group of users. Users can send database files, input streams, spooled output files, messages, and save files using the system distribution directory (for local users), or a SNA distribution services (SNADS) network and the system distribution directory (for remote users).

When sending information using object distribution, you can specify the recipient of the information using the SNDNETF, SNDNETMSG, SNDNETSPLF, or SBMNETJOB commands. You can specify either an individual user by user ID and address or a group of users by a distribution list. The following list shows what the commands do:

| | |
|-------------------|---------------------------|
| SNDNETF | Send Network File |
| SNDNETMSG | Send Network Message |
| SNDNETSPLF | Send Network Spooled File |
| SBMNETJOB | Submit Network Job |

Use the Work with Network Files (WRKNETF) command to work with files on the network.

User Restrictions

The system distribution directory has these restrictions:

- The user ID and address together must be unique in the directory and in the shadowing network. If your system is part of a SNADS network, you should also make the user ID and address pair unique in the network.
- The network user ID must be unique in the directory and shadowing network. The network user ID takes the user ID and address as its default value but can be changed to any value you specify. It is used to synchronize directory entries in shadowing.
- The description must be unique within a user ID and address.
Note: This restriction applies only when a user ID and address has multiple descriptions. Each description must be unique within that user ID and address.
- A user profile field in the directory entry cannot be changed if the user is enrolled in OfficeVision/400.
- For compatibility with the VM/MVS bridge, the user address should be named the same as the local system. The VM/MVS bridge requires this.
Note: If compatibility with the VM/MVS bridge is not required, the user address can be any value you choose. For example, this allows two or more local users to have the same user ID with different addresses.
- If you do not have security administration special authority or higher, you can view the entire directory, but you cannot add, change, or remove other users. You can change specific fields in your own entry, but you are not allowed to remove or add your own entry in the directory.

Distribution lists have the following restrictions:

- The list ID must be unique among distribution lists or user IDs and addresses.
- The creator of a distribution list becomes the owner of that list. Only the owner, administrator (*SECADM special authority), or a security officer can change or delete a distribution list.

There is no limit to either the number of distribution lists a person can own, or the number of entries in a specific list.

Security Administrator Restrictions

The **security administrator (*SECADM) special authority** allows a user to perform office administration functions for enrollment and management of the document library. Special authorities may be assigned to users with the SPCAUT parameter on the CRTUSRPRF or CHGUSRPRF commands. The *SECADM special authority gives the user authority to use commands, specific options on commands, or use commands to manage objects without explicit authorization to the objects. See the security considerations of each command for more information.

IBM Profile Directory Entries

The system distribution directory is an IBM-supplied object. You must determine which users to add to the directory. It is shipped with the following entries. Only the entries for QSECOFR and QUSER can be renamed.

- | | | |
|----------------|-----------------|--|
| | QDFTOWN | The default entry is used as the owner of a folder, document, or distribution list when the originating owner is no longer in the directory. This entry cannot be removed from the directory, and it is recommended that distributions not be sent to this user. |
| | QSYS | An internal system entry that owns IBM-supplied folders and documents shipped with the system. This entry cannot be removed from the directory, and it is recommended that distributions not be sent to this user. |
| | QSECOFR | The system security officer entry used by some licensed programs when they are installed. This entry can be removed, but it is suggested that an entry be kept in the directory for the security officer unless you require that the security officer not be enrolled on your system. If you want a different user ID and address, replace the original entry with a new one. The same rules exist for removal, so if the security officer owns a document, the entry cannot be deleted. |
| | QLPINSTL | An internal system entry used by the licensed program installation function. This entry cannot be removed from the directory, and it is recommended that distributions not be sent to this user. |
| | QLPAUTO | An internal system entry used by the automatic licensed program installation function. This entry cannot be removed from the directory. It is recommended that distributions not be sent to this user. |
| | QUSER | The entry used by the Operating System/2* (OS/2*) version of PC Support/400 shared folders when the system security level is 10. This entry can be removed if the system security level is not 10 or if PC Support/400 is not being used. It is recommended that distributions not be sent to this user. This entry can be renamed, but it is not suggested. |
| | QDOC | An internal system entry used to own IBM-supplied libraries, files, journals, and programs. This entry cannot be removed from the directory. It is recommended that distributions not be sent to this user. |

Directory Services Control Language Interfaces

This section describes the system distribution directory commands. Each command description discusses the use of the command, the security considerations for the command, and any other unique considerations. The distribution list commands are described in “Distribution List Commands” on page 3-19.

The directory services commands are:

- Add Directory Entry (ADDDIRE)
- Change Directory Attributes (CHGDIRA)
- Change Directory Entry (CHGDIRE)
- Display Directory (DSPDIR)
- Remove Directory Entry (RMVDIRE)
- Rename Directory Entry (RNMDIRE)
- Work With Directory (WRKDIR)

Add Directory Entry (ADDDIRE) Command

The Add Directory Entry (ADDDIRE) command adds new entries to the system distribution directory. The directory contains information about a user, such as the user ID and address, system name, user profile name, mailing address, telephone number, and other text information. There are also fields for name, department, job title, company, location, office, building, and X.400 originator/recipient name. The ADDDIRE command provides a parameter for each of the fields contained in the directory.

Security Considerations

You must have security administrator or security officer authority to use this command.

Some Ways to Use This Command

The following describe some ways to use this command:

- Much of the directory data you need may already exist in other files (for example, mailing addresses and telephone numbers). You can write programs to extract the data from these existing files and to add this data to the directory using the ADDDIRE command.

Adding a local user to the directory requires that the user profile already exist on the system. Therefore, it is not possible to load the directory from a personnel file using only the directory services commands. Also, it is unlikely that the user ID and address exists within a customer file. An application may extract the data for one user, present a display to the user requesting the user ID/address and profile, and then create the user profile, using the Create User Profile (CRTUSRPRF) command, and add the directory entry.

- You may also use this command to add a user to the system distribution directory when adding an employee to your personnel file. Include the ADDDIRE command in your program and bring up a display asking if this person should be added to the system distribution directory.
- To include all the users already enrolled on your system (user profiles) in the system distribution directory, you can write a program to use existing user profile (USRPRF) data to create new directory entries. The user profile data can be placed in a database file using the OUTFILE parameter of the

DSPOBJD command, as shown below. Your program can then read that data and use it to create new data entries.

For the user description, use the text from the user profile (ONLY if your user profile text is UNIQUE and NOT BLANK). Or, get the description and other information such as address and telephone number from a personnel file.

It is desirable to have the user profile name and user ID the same, but the user profile is 10 characters in length and the user ID is only 8 characters in length. If you allow more than 8 characters for the user profile, you need to decide how to handle this situation (truncating is one solution).

Use the following command to write the user profiles to an output file:

```
DSPOBJD OBJ(QSYS/*ALL) OBJTYPE(*USRPRF) OUTPUT(*OUTFILE)
        OUTFILE(library-name/file-name)
```

You can add only the new users by using the ADDDIRE command. If an entry for the user profile already exists, the error message CPD8981 appears.

Rules

The following rules for the user ID and address apply to the ADDDIRE command:

- The user ID and address consist of 1 to 8 characters each.
- Lowercase characters are converted to uppercase.
- The first character of either a user ID or address cannot be blank.
- Blanks can be embedded in either the user ID or address.
- Trailing blanks are ignored.
- See the *Distribution Services Network Guide* for characters allowed for the user ID and address.

You cannot use the ADDDIRE command to add multiple descriptions to a user. Use the WRKDIR command to do this.

Hints and Techniques

The following hints and techniques will help you use this command:

- Make sure the descriptions are consistent with other descriptions when added to the directory. For example, use Last name, First name as the first description of a user. This helps when sorting by description using the WRKDIR command.
- Use the WRKDIR command to add directory entries interactively.

Character Set and Code Page Considerations

The USRID, SYSNAME, DEPT, and LOC parameter values are translated from the value specified in the CMDCHRID parameter to character set 697 and code page 500. The X.400 originator/recipient name is translated to the International Alphabet No. 5 (IA5). No other fields are translated. They are kept in the directory with the character set and code page as specified in the CMDCHRID parameter when entered in batch, or the character set and code page when entered interactively.

Change Directory Attributes (CHGDIRA) Command

The Change Directory Attributes (CHGDIRA) command changes directory attributes when working with the directory and directory shadowing. You can change the following directory attributes:

| Function | Attributes |
|-------------------------------|--|
| Directory search | Search type Search user exit program |
| System distribution directory | Allow display of network user ID |
| Directory shadowing | Retry interval Retry limit Check authority user exit program Supplier user exit program Message queue Shadow remote users |

Security Considerations

You must have security administrator (*SECADM) or all object (*ALLOBJ) special authority to change the directory attributes. To change the search or the check authority user exit program, *ALLOBJ special authority is required.

Some Ways to Use this Command

Use the CHGDIRA command for the following:

- If you want to do a generic search on the directory, you normally need to put an asterisk (*) at the end of the search string. By specifying SCTYPE(*GENERIC) on the CHGDIRA command, the search becomes an automatic generic search and you do not need to use an asterisk at the end of the search string.
- If you need a field in the directory entry that contains confidential information, such as a user's social security number, you can protect it from being displayed to the public. Place the value in the network user ID field and specify ALWDSPNUI(*NO) on the CHGDIRA command. This allows the field to be displayed only to system administrators or users displaying their own information. This field must be unique on the system and in the network for shadowing.
- You can set directory shadowing values, such as the number of times a directory shadow should be retried before failing and the time interval between shadowing attempts.

Change Directory Entry (CHGDIRE) Command

The Change Directory Entry (CHGDIRE) command changes the data for an existing entry in the system distribution directory, such as the entry's system name and group, indirect user status, mailing address, location, telephone number, and other text information.

Security Considerations

You must have security administrator (*SECADM) or all object (*ALLOBJ) authority to change the data in a directory entry for any user on the system.

Users that are not administrators are restricted to updating their own directory entry and have a limited set of fields in the entry (address, location, telephone numbers, and text) that can be updated.

Some Ways to Use This Command

Use the CHGDIRE command for the following:

- If data in a personnel file changes, use the CHGDIRE command to have your application program update the directory at the same time.
- If you have remote users specified in your directory, use an output file from the DSPDIR command from the remote system and compare the remote users on your system with that file. If they have different information, use the CHGDIRE command to update and synchronize the data. (Use the ADDDIRE command to add new entries and the RMVDIRE command to delete obsolete entries.)

Rules

You cannot change the user ID or address with this command. Use the Rename Directory Entry (RNMDIRE) command or option 7 (Rename) on the Work with Directory (WRKDIR) display to change these values.

Hints and Techniques

To change directory entries interactively, use the WRKDIR command.

Character Set and Code Page Considerations

Only the fields being changed reflect the current character set and code page value. For example, if the directory entry was originally added from an English work station, the fields in the entry would be in English. If you change ADDR1 and TEXT from a German work station, only the ADDR1 and TEXT fields would reflect the German CHRID. When the SYSNAME, DEPT, or LOC values are changed, the CMDCHRID value is used to translate these to character set 697 and code page 500.

Display Directory (DSPDIR) Command

The Display Directory (DSPDIR) command displays some or all directory entries, both local and shadowed. The command displays the entries at the requesting work station, prints them with the job's spooled output on a printer, or adds them to a database file.

Security Considerations

No special authority is needed for the DSPDIR command.

Some Ways to Use This Command

This command allows data to be sent to an output file. When this is done, the output file can be accessed.

When run interactively, the display from the DSPDIR command has an interface to display the user's nicknames, display the departments defined in the directory, and to search the directory. The search allows you to specify a user's name (first, last, middle, preferred), department, user ID, user address, location, network user ID,

building, telephone number, and company. A display returns showing all directory entries that meet the search value.

Use the DSPDIR command to:

- Look at the number of items in each user's mail log in the system distribution directory. To do this, use DSPDIR *ALL and send the output to an output file. Do a QRYDST command on each record in the output file. If there are too many, send a message to that effect. If there are old messages and notes, send a warning message that you will delete the notes if they do not do so. If necessary, you can delete those items using the DLTDST command. (See "Maintaining Mail Logs" on page 4-73 for an example.)
- Search the directory for telephone numbers. If the search function does not provide what you want, you can provide your own telephone lookup command. Use the DSPDIR command, send the list to an output file, and then search for the name specified. This works well if you used the Last Name, First name format for the description. You could create a logical file over this file to order it, giving you better performance on the search for the name.

Rules

The following rules apply to the DSPDIR command:

- When there is more than one entry to be displayed interactively, the system provides a list of entries. You then can choose an option to display or to print the details of one or more of the directory entries.
- When the DSPDIR parameters uniquely identify a directory entry, the system displays the details for the entry.
- When directing output to an output file, the format of the file is created by the system from the model output file QAOSDIRO, QAOSDIRB, QAOSDIRF, or QAOSDIRX in library QSYS. See "Output File Layout for DSPDIR" for more information. If the file already exists, you can specify *REPLACE or *ADD. If the file exists and the format is wrong, the system sends an error message.

Output File Layout for DSPDIR

You can specify three types of output file formats using the `OUTFILFMT` parameter. (See "Directory Services Overview" on page 3-1 for an explanation of these output file formats.) The model output file supplied with the system is QAOSDIRO in library QSYS when `OUTFILFMT(*TYPE1)` is specified. When `OUTFILFMT(*TYPE1)` is specified with `DETAIL(*BASIC)`, only the fields WOSDDEN, WOSDDGN, and WOSDDESC are written to this file.

`OUTFILFMT(*TYPE2)` with `DETAIL(*BASIC)` specified generates an output file of model format QAOSDIRB. This output file has the basic values. `OUTFILFMT(*TYPE3)` and `DETAIL(*BASIC)` also generates this file.

`OUTFILFMT(*TYPE2)` with `DETAIL(*FULL)` specified generates an output file of model format QAOSDIRF. This has all directory fields, except X.400 fields.

`OUTFILFMT(*TYPE3)` with `DETAIL(*FULL)` specified generates an output file of model format QAOSDIRX. This has all the directory fields including the X.400 originator/recipient name fields.

If you are using `OUTFILFMT(*TYPE1)`, the field names are longer than 6 characters, so they must be renamed if used in RPG/400* programs. If you are using

OUTFILFMT(*TYPE2) or OUTFILFMT(*TYPE3), these generate output files with field names of 6 characters or less.

A record is written for each system directory entry as selected in the user ID parameter.

Note: When multiple descriptions exist in the directory for a user ID and address, one record is written for each description. The only data that is different between these records is the description field.

For more detailed information about the output file layout, see Appendix C, "Office Services Output File Considerations."

Hints and Techniques

The DSPDIR command does not allow updating to the directory. That function is provided by the ADDDIRE, CHGDIRE, RMVDIRE, RNMDIRE, or WRKDIR commands. With the WRKDIR command, the directory entries are displayed and you can update the directory interactively.

Character Set and Code Page Considerations

USRID is translated from the value specified in CMDCHRID parameter to character set 930 and code page 500. None of the other fields are translated.

The CHRID for the user ID is not sent to the output file.

Remove Directory Entry (RMVDIRE) Command

The Remove Directory Entry (RMVDIRE) command removes a specific entry from the system distribution directory. The user ID and address are also removed from any distribution lists. If a user ID and address have multiple descriptions, you may choose to remove only a specific description or all descriptions.

Security Considerations

You must have security administrator or security officer authority to remove a directory entry with this command.

Some Ways to Use This Command

The following describe some ways to use this command:

- You can use this command to remove a user from the system distribution directory. To do this, delete or change the owner of all current documents and folders, delete their distributions, and remove them from OfficeVision/400 (if enrolled). Then you can remove the user from the system distribution directory.

Write an application program that changes the owner of the folders and documents using the CHGDLOOWN OWNER(user-profile-name) NEWOWN(user-profile-name) command. Or you may want to delete all documents and folders this user owns by using DLTDLO DLO(*SEARCH) FLR(*ANY) CRTDATE>(*AVAIL *BEGIN) (*AVAIL) OWNER(user-profile-name). To delete the user's distribution, use QRYDST and from the output file use DLTDST to delete the distributions. After this is done, use the RMVDIRE command to remove the user from the system distribution directory.

- If you do not want to have specific entries for remote users, you may want to remove any user who moves from the local system to a remote system. Then you can add them in the same way you add other remote users.

Rules

You cannot remove a user if:

- The user owns documents and folders
- The user is enrolled in OfficeVision/400
- The user has mail

When a user is removed from the system distribution directory, all references to that user in distribution lists are deleted unless you are removing an *ANY directory entry. If an *ANY *ANY directory entry exists, the distribution list entries remain, but the description changes to the description of the *ANY *ANY entry.

If multiple descriptions exist, you can remove the first entry, all the entries, or an entry with a specified description. When removing an entry using the description, the description must match the description in the system exactly before it can be removed (uppercase/lowercase is significant).

Hints and Techniques

Make sure the user does not own any documents or folders, is not enrolled in OfficeVision/400, and has removed all mail from the distribution recipient index before attempting to remove the user from the directory.

Character Set and Code Page Considerations

USRID parameter value is translated from the value specified in the CMDCHRID parameter to character set 930 and code page 500. None of the other fields are translated.

Rename Directory Entry (RNMDIRE) Command

The Rename Directory Entry (RNMDIRE) command renames a local or remote user ID and address to a new user ID and address. All occurrences of this user ID and address in all the IBM-supplied files are renamed to the new user ID and address.

Local users can have distributions automatically forwarded from the old user ID and address or a specified user ID and address. You can also have the network user ID and address changed to the new user ID and address at the time the RNMDIRE command is run.

Changing the address part of the user ID and address does not change the system name of the directory entry. To route the distributions for the user to a system other than the one specified by the directory entry, use the CHGDIRE CL command.

Security Considerations

You must have security administrator (*SECADM) or all object (*ALLOBJ) special authority to rename a user ID and address.

Rules

Generic (*ANY) user IDs cannot be renamed.

Renaming the default directory entries QDFTOWN, QSYS, QLPINSTL, QLPAUTO, and QDOC is not allowed.

There cannot be more than one RNMDIRE command running on the system at one time. If a RNMDIRE command is submitted to batch, the job waits for any active RNMDIRE command to complete.

No office activity is allowed during the renaming operation of a local user. (Office activity includes using OfficeVision/400, PC Support/400, the hierarchical file system (HFS), SNADS, and Transmission Control Protocol/Internet Protocol (TCP/IP)). The QSNADS subsystem and the job QTMSMTP running under the QTCP subsystem must be ended before renaming user IDs and addresses.

No OfficeVision/400 calendar, system distribution, or QTMSMTP activity is allowed during the renaming operation of a remote user.

Automatic cleanup through operational assistance cannot run during the rename operation. Specify CHGCLNUP ALWCLNUP(*NO) to stop automatic cleanup before the RNMDIRE command is run.

Hints and Techniques

For local users, a rename operation is a long-running operation because many changes to the system are involved. The RNMDIRE command does not allow office or calendar activity to occur while it is running. It is recommended that running the RNMDIRE command be a scheduled job. This can be done by specifying RNMDIRE on the SBMJOB command or by using option 7 (Rename) from the Work with Directory (WRKDIR) display and submitting it to batch.

Character Set and Code Page Considerations

The user ID and address parameters are translated from the job CCSID value to character set 930 and code page 500. To change the job CCSID value, use the Change Job (CHGJOB) command.

Work with Directory (WRKDIR) Command

The Work with Directory (WRKDIR) command provides a set of displays that allow a user to add, change, rename, display, and delete entries in the system distribution directory. The system distribution directory contains information about local, remote, or indirect users used to distribute electronic mail. When this command is used, the system shows one or all of the entries in the system directory, depending on the kind of user (security officer, administrator, or regular user) and the parameters specified (user ID, address, user profile).

Nicknames can be managed through the WRKDIR or WRKDSTL commands. You can have a nickname for a user or for a distribution list.

You can manage department information through the WRKDIR or WRKDSTL command. There is also an interface from the WRKDIR and WRKDSTL commands to search the directory.

You can manage locations through the WRKDIR or WRKDIRLOC command.

Security Considerations

If you are not an administrator, you can change only a restricted set of fields in your own directory. If you are a security administrator or security officer, then a list of all users is shown when USRID(*ALL) is used. Otherwise, the specific directory entry is shown depending on the parameter specified.

Some Ways to Use This Command

Use the WRKDIR command to:

- Work with directory entries interactively.
- Add, remove, or change a directory entry interactively with easy-to-use interfaces.
- View and change entries in the directory without entering additional commands. Simply choose the option from the display.
- Create, view, change, print, and delete nicknames, location, and department information.
- Search the directory.
- Rename a directory entry.

Rules

This command only can be used interactively. The same rules and restrictions apply as described under the commands for adding a user (ADDDIRE), changing an entry (CHGDIRE), removing an entry (RMVDIRE), and renaming an entry (RNMDIRE).

Hints and Techniques

Use this command for managing your system distribution when working interactively with the system. Generally, the specific directory commands are best used in programs.

Use F17 to position to the user you want to work with (instead of paging through the directory). Press the Sort by description (F11) key to allow F17 to position to the description of the user you want to work with.

WRKDIR is the only interface for adding multiple descriptions to a user ID and for assigning a different ID to a description.

WRKDIR is the easiest way to remove one of multiple descriptions for a user ID. It can be done with RMVDIRE, but the value specified in the description parameter must match exactly with the directory value for the remove to work.

You cannot use nicknames with any of the CL commands. You can use nicknames within the panels displayed by the WRKDIR and WRKDSTL commands and through OfficeVision/400. For example, you cannot send a note using a nickname with the SNDDST command, but you can send a note using the OfficeVision/400 send mail function.

Character Set and Code Page Considerations

The user ID and address are translated from the value specified in the CMDCHRID parameter to character set 930 and code page 500. No other parameter values are translated.

Distribution List Commands

A distribution list is a collection of user IDs and addresses. It is used to send information to a group of users. A distribution list can include local users, remote users, or remote distribution lists.

The following section describes each distribution list command individually and discusses some ways to use each command, the security considerations for the command, and any unique considerations. The following distribution list commands are described:

- Add Distribution List Entry (ADDDSTLE)
- Create Distribution List (CRTDSTL)
- Delete Distribution List (DLTDSTL)
- Display Distribution List (DSPDSTL)
- Remove Distribution List Entry (RMVDSTLE)
- Work with Distribution List (WRKDSTL)

Add Distribution List Entry (ADDDSTLE) Command

The Add Distribution List Entry (ADDDSTLE) command adds new entries to an existing distribution list. A distribution list can include local, remote, and indirect users.

A distribution list can also include remote distribution lists but not local distribution lists. A remote distribution list is a local entry in the system distribution directory, but on the remote system it is actually a distribution list.

Security Considerations

You must be a security administrator or a security officer to add entries to a distribution list owned by someone else.

Users can add entries to a distribution list they have created without security restrictions.

Some Ways to Use This Command

Use the ADDDSTLE command to make distribution lists for each department in your organization from a personnel file with the department number in each record. Read the personnel file and use the ADDDSTLE command to add the user to each department distribution list.

Rules

The distribution list must exist before entries can be added to it. Use the CRTDSTL command to create the distribution list.

Local distribution lists cannot be included as an entry in distribution lists, but can be used to copy their entries into the list.

Up to 300 entries can be added to a distribution list at one time. In addition, up to 50 local distribution list IDs can be specified, whose entries are copied into the distribution list.

Duplicate entries can exist in a distribution list. Document distribution services ignore duplicate entries.

When a local distribution list is copied to another distribution list, the entries are copied. If changes are made to one distribution list, they are not automatically changed in the other distribution list.

When an entry is being added, and the specific user ID and address are not found in the directory, the directory is searched for an entry with user ID *ANY and an address that matches the specified address. If this is found, the user ID and address are added to the distribution list. If it is not found, but an *ANY *ANY entry exists, the user ID and address are added to the distribution list. If no *ANY entry is found, the entry is not added to the list and an error message is returned.

Hints and Techniques

To copy one distribution list to another, use the FROMLSTID parameter. Use the USRID parameter to add specific users.

When copying distribution lists to your distribution list, a message indicates how many entries were added. This number reflects the number of distribution lists and specific entries added. For example, if you add four users and two distribution lists and they are successful, the message returns with six entries added to the distribution list even though the two distribution lists had multiple users in them.

Character Set and Code Page Considerations

USRID, LSTID, and FROMLSTID parameter values are translated from the value specified in the CMDCHRID parameter to character set 930 and code page 500.

Create Distribution List (CRTDSTL) Command

The Create Distribution List (CRTDSTL) command creates a new distribution list. A distribution list is a list of entries from the system distribution directory. The CRTDSTL command simply creates the distribution list with no entries. Entries can be added with the ADDDSTLE command.

Security Considerations

You must be enrolled in the system directory to create a distribution list. After the distribution list is created, only the person who created it, a security officer, or administrator can add to it, remove entries from it, or delete it. Anyone can view this distribution list.

Some Ways to Use This Command

Use the CRTDSTL command to create multiple distribution lists based on information that exists in the system directory. A program can be written to use the information to create distribution lists, such as department distribution lists.

Rules

The list ID is a two-part identifier that has the same rules as the user ID and address (such as number of characters, allowed characters, and so on). See the “Add Directory Entry (ADDIRE) Command” on page 3-10 for these rules.

The description can be 1 to 50 characters in length. The description is not translated and has no restrictions on what it can contain.

The list ID must be different from:

- Any other list ID
- Any other user ID and address
- Any other entry in a distribution list

Hints and Techniques

The list description is used only as a reference and does not have to be unique. If you want to write an application using the description of a distribution list (like sorting by description), the description should be meaningful for your application.

Once the distribution list is created with a description, the description cannot be changed.

Character Set and Code Page Considerations

The LSTID parameter value is translated from the value specified in the CMDCHRID parameter to character set 930 and code page 500. The list description is not translated.

Delete Distribution List (DLTDSTL) Command

The Delete Distribution List (DLTDSTL) command deletes an existing distribution list.

Security Considerations

Only administrators and security officers can delete a distribution list owned by another user. Users can delete distribution lists they created without restrictions.

Some Ways to Use This Command

The following describe some ways to use the DLTDSTL command:

- You may have empty distribution lists, but if you want to clean them up, do a DSPDSTL command to an output file with option DETAIL(*FULL). Look for records that have only a list header record without list entry records under them, and then do a DLTDSTL command for those distribution lists with only a header record.
- If you have an application that periodically goes out and re-creates distribution lists in order to update them, use this command to delete them.

Rules

If a specified list identifier does not exist in the directory, an error message is returned. If multiple list identifiers are given, each one is processed. An error message is returned each time a list identifier does not exist or when the current user of this command does not have authority to delete the distribution list.

One to 300 distribution lists can be deleted at one time.

Character Set and Code Page Considerations

The LSTID parameter value is translated from the value specified in the CMDCHRID parameter to character set 930 and code page 500.

Display Distribution List (DSPDSTL) Command

The Display Distribution List (DSPDSTL) command is used to display a distribution list or a list of all distribution lists at a requesting work station, print them, or add them to a output file. If output is to a printed list or a data file, the DETAIL parameter determines whether a list of distribution lists or the details of who is in each distribution list are the output. You can view the departments and nicknames, when you run the command interactively.

Security Considerations

No special authority is needed to display distribution lists on the system.

Some Ways to Use This Command

Use the DSPDSTL command to:

- Look at all of your distribution lists on your system interactively. From the list of distribution lists, you can list the entries within a distribution list or print a distribution list.
- Create a database output file of one or all distribution lists. Your application program can access the fields of this output file. The layout of this output file is discussed later in this section.

List all the distribution lists in the directory by using the option DETAIL(*BASIC). If the option DETAIL(*FULL) is specified, you get a list of all the distribution lists in the directory and all the entries in each distribution list.

- Search the output file of the DSPDSTL command for the distribution lists that do not have any entries and delete them. This is explained in more detail in “Delete Distribution List (DLTDSTL) Command” on page 3-22.
- Use the description of the distribution list to sort, find a distribution list, and so on. To do this, use the DSPDSTL command to an output file with the option DETAIL(*BASIC). This lists all the distribution lists in the directory but not the entries of each distribution list.
- Update remote distribution list entries on remote systems. Send the output file from the DSPDSTL command to the remote system, and run an application program to update the directory with the new remote distribution lists.

Rules

If the output file already exists, it needs to be in the same format as QAOSDSTO in library QSYS, record format name OSDSTL.

When LSTID is specified, both the list ID and the list qualifier need to be specified.

If the specified list identifier does not exist in the directory, an error message appears.

Output File Layout for DSPDSTL

The model output file supplied with the system is QAOSDSTO in library QSYS.

Since the field names are longer than 6 characters, they have to be renamed if used in RPG/400 programs.

If DETAIL(*BASIC) was selected, a record is written for every distribution list in the directory. If DETAIL(*FULL) was selected, a header record is written for the distri-

bution list (record code 0). Following the header record for each distribution list are the distribution list entries (record code 1).

For more detailed information about these fields in the output file, refer to Appendix C, "Office Services Output File Considerations."

Hints and Techniques

There are two types of lists sent to the output file depending on whether you choose `DETAIL(*BASIC)` or `DETAIL(*FULL)`. If your application is working with the distribution lists and not the entries, use `DETAIL(*BASIC)`. If you need the entries of every distribution list on the system, use `DETAIL(*FULL)`.

If you want to see a list of distribution lists and work with them, use the `WRKDSTL` command. From the `WRKDSTL` display you can change entries, delete lists, display entries, and print entries. From the `DSPDSTL` display you can only display and print entries.

Character Set and Code Page Considerations

The `LSTID` parameter value is translated from the value specified in the `CMDCHRID` parameter to character set 930 and code page 500.

Remove Distribution List Entry (RMVDSTLE) Command

The Remove Distribution List Entry (`RMVDSTLE`) command removes entries from a distribution list. Up to 300 sets of user IDs, addresses, and descriptions can be specified at one time. Each valid set is removed from the distribution list.

Security Considerations

You must have security administrator or security officer authority to remove entries from a distribution list owned by someone else. You may remove entries from a distribution list that you created without any special authorities.

Some Ways to Use This Command

Use the `RMVDSTLE` command to take a user ID out of your distribution lists. Use the `DSPDSTL` command to create an output file with `DETAIL(*FULL)`. Search the output file for the user ID (field `WOSLEID`) to be removed. Use the `RMVDSTLE` command with the field `WOSLLST` for the `LSTID` parameter and field `WOSLEID` for the `USRID` parameter. (You must split up the `WOSLLST` and `WOSLEID` fields to have a two-part identification for the parameter.)

Rules

Up to 300 sets of user IDs can be specified at one time for one distribution list.

When `LSTID` is specified, both the list ID and the list qualifier must be specified.

When `USRID` is specified, both the user ID and address must be specified.

If you specify the user-description when removing a user ID, the specified description must match (uppercase and lowercase included) the description exactly as it exists within the list. If they are not an exact match, the entry is not removed and a message is displayed.

Hints and Techniques

To ensure that all entries for a user ID are removed from a distribution list, you must specify USRID (user ID address *ALL). The default is to remove the first occurrence of the user ID and address.

Do not specify the user-description in the USRID parameter, if possible. It is too easy to specify it incorrectly. Instead use *FIRST or *ALL.

Character Set and Code Page Considerations

The LSTID and USRID parameter values are translated from the value specified in the CMDCHRID parameter to character set 930 and code page 500.

The user-description in the USRID parameter is not translated. If specified, the user-description is in the character set and code page as specified in the CMDCHRID parameter if in batch, or in the terminal character set code page if interactive. If the distribution list entry was added to the distribution list in a different character set and code page than specified by the RMVDSTLE command, they may not match.

Work with Distribution List (WRKDSTL) Command

The Work with Distribution List (WRKDSTL) command provides a set of displays to add, remove, display, and print entries as well as create and delete lists. When this command is used, the system shows one or all distribution lists, depending on the value specified in the LSTID parameter.

Nicknames are supported by this command. You can work with nicknames by pressing F9. Using this command you can create, view, change, print, or delete nicknames. You can also use the WRKDIR command to work with nicknames and departments if you have *SECADM authority.

Security Considerations

You do not need special authority to display or print the entries of a distribution list. To change entries in a list, or delete a list, the user must be the owner or have security officer or security administrator authority.

Some Ways to Use This Command

To work with several distribution lists at one time interactively, the WRKDSTL command is the best choice. Also, to change entries, delete lists, display entries, or print entries interactively, this command has a better user interface than the ADDDSTLE, DLTDSTL, DSPDSTLE, RMVDSTLE, and CRTDSTL CL commands.

This command also works for general displaying and changing items in a distribution list, without entering another command. You can choose the option from the display.

Rules

This command can only be used interactively.

The same rules and restrictions apply as for ADDDSTLE, DLTDSTL, DSPDSTL, RMVDSTLE, and CRTDSTL.

Hints and Techniques

Use this command for managing your distribution lists or for working interactively with the system. Generally, only use the specific distribution commands in programs.

Use the F17 key to position to the distribution list you want to work with (instead of paging through the lists).

WRKDSTL and WRKDIR are the only CL commands to work with nicknames. You cannot use nicknames with any of the other CL commands, but you can use nicknames through OfficeVision/400. For example, you cannot send a note using a nickname with the SNDDST command, but you can by using OfficeVision/400.

Character Set and Code Page Considerations

The LSTID parameter value is translated from the value specified in the CMDCHRID parameter to character set 930 and code page 500.

Example Applications

The following examples show how you can use the CL commands for distribution directory management and updating of the directory from customer data.

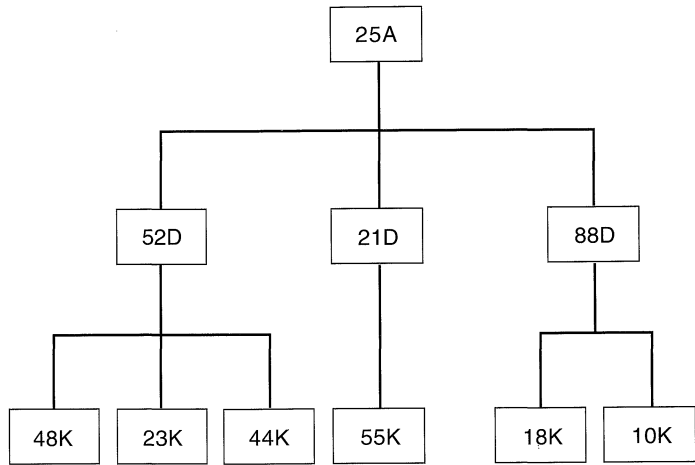
Maintaining Multiple Distribution Lists

Because distribution lists cannot contain local distribution lists, you may want to write an application program that helps you maintain some distribution lists that you normally include in other lists that are changed frequently.

An example is a department list in an organization. If your company has four levels, you may want distribution lists at each of those levels. People change departments so you may want a program that updates the distribution lists accordingly.

Assume a file is set up with the line organization. As Figure 3-1 on page 3-27 shows, a record code that starts with 3 defines the organization's primary departments and the record code that starts with 1 defines the top department in the organization.

Organization Chart



RSLW129-0

File name: LINEFILE

```
3 48K
3 23K
3 44K
2 52D
3 55K
2 21D
3 18K
3 10K
2 88D
1 25A
```

Figure 3-1. Line Organization Diagram

With the LINEFILE layout you could then write a program to update your organization distribution lists.

The program UPDORGLST takes as input a user ID and address, the department number, and the option of either to add the user or remove the user. The program processes each department distribution list in the LINEFILE file and adds or removes (depending on the option sent) the user to or from the distribution lists in that user's department line.

The DDS for the LINEFILE is:

```
A** DDS OBJECT:      LINEFILE
A** SOURCE FILE:    QDDSSRC
A** SOURCE LIB:     OFCLIB
A** DESCRIPTION:    LINE FILE FOR ORGANIZATION
A*
A          R DEPTRCD
A          RCD              2  0
A          LINDPT          3A
```

The UPDORGLST CL program is the following:

```

PGM          PARM(&USRID &ADDR &DEPT &OPTION)
/* Program name: UPDORGLST */
/* This program will update the organization's */
/* distribution lists to reflect the new or old */
/* department number input. */
/* */
/* Dependency: The department distribution lists */
/* must already exist before this */
/* program is called. */
/* These names are assumed to be */
/* DEPTnnn where nnn is the dept number */
/* and SYS1 is the list ID qualifier. */
/* */
/* Input: */
/* LINEFILE file */
/* User ID (&USRID) and address (&ADDR) */
/* Department number */
/* Option to add or remove from a department */
/* (*ADD or *RMV) */
/* */
/* Example call of this program from the command */
/* line would be: */
/* CALL UPDORGLST PARM(JOHN SYS1 48K *ADD) */
/* */
/* where JOHN is the user ID at address SYS1 */
/* and he is added to department 48K. */
/* */
/* To compile this program use the CRTCLPGM command. */
/* */
DCL          VAR(&USRID) TYPE(*CHAR) /* Variable +
           for the User ID parameter */
DCL          VAR(&ADDR) TYPE(*CHAR) /* Variable +
           for the Address parameter */
DCL          VAR(&DEPT) TYPE(*CHAR) LEN(3) /* +
           Variable for department number */
DCL          VAR(&OPTION) TYPE(*CHAR) LEN(4) /* +
           Option to add or remove from a dept */
DCL          VAR(&CATDEPT) TYPE(*CHAR) LEN(7) /* +
           Concatenated department name */
DCL          VAR(&CNTRCD) TYPE(*DEC) LEN(2 0) /* +
           Count of the record code */
DCLF        FILE(OFCLIB/LINEFILE) /* File +
           containing organizations departments */

/* Search the LINEFILE file for the department number */
/* input. When found, process each department in */
/* the hierarchy and add or remove the user from the */
/* department distribution lists depending on the */
/* option sent. */

READF:      RCVF RCDFMT(DEPTRCD)
           /* Monitor for EOF message */
MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

```

Figure 3-2 (Part 1 of 2). UPDORGLST CL Program

```

/* If the department specified is found, then save */
/* the record code and go to process the rest of the */
/* departments in the hierarchy. */
IF (&DEPT *EQ &LINDPT) (DO)
    CHGVAR VAR(&CNTRCD) VALUE(&RCD)
    GOTO PROCESS
ENDDO

GOTO CMDLBL(READF)

/* The distribution list is 'DEPT' concatenated with */
/* the department number in the record. The list ID */
/* qualifier is 'SYS1'. Change this to meet your */
/* needs. */
PROCESS: CHGVAR VAR(&CATDEPT) VALUE('DEPT' *CAT &LINDPT)
/* If the option passed is to add to the department */
/* then add the user to the department's distribution */
/* list, else remove it. (This code is assuming if */
/* the option is not *ADD it is *RMV). */
IF (&OPTION *EQ '*ADD') THEN( +
DO)
    ADDDSTLE LSTID(&CATDEPT SYS1) USRID((&USRID &ADDR))
/* Monitor for user not in distribution list */
    MONMSG MSGID(CPF9091) EXEC(GOTO CMDLBL(HNDLERR))
ENDDO
ELSE +
DO
    RMDVSTLE LSTID(&CATDEPT SYS1) USRID((&USRID &ADDR))
/* Monitor for user not in distribution list */
    MONMSG MSGID(CPF9093) EXEC(GOTO CMDLBL(NEXT))
ENDDO

/* Change the variable to one less than the record */
/* code found so that the next department in the */
/* hierarchy can be found. */
NEXT: CHGVAR VAR(&CNTRCD) VALUE(&CNTRCD - 1)

/* Read the records in the LINEFILE file until the */
/* record code is found of the next department in */
/* the hierarchy. When it is found, process it. */
READ2: RCVF RCDFMT(DEPTRCD)
/* Monitor for EOF message */
MONMSG MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
IF (&RCD *EQ &CNTRCD) (GOTO PROCESS)
GOTO READ2

HNDLERR: CHGVAR VAR(&OPTION) VALUE(*ERR)

EOF:
END: RETURN /* Return control to the calling program. */
ENDPGM

```

Figure 3-2 (Part 2 of 2). UPDORGLST CL Program

The department information is stored in the directory. If the department fields for all users have been updated, the following program uses the department information from the directory.

This example shows how to re-create all of your organization's department lists from the LINEFILE file.

You may want to distribute the new organization distribution lists to remote systems after you have processed them on one of your systems.

The CRTORGLST CL program is the following:

```
PGM
/* Program name: CRTORGLST */
/* This program will create all the organization's */
/* department distribution lists from your personnel */
/* file. The name of the list is DEPTnnn SYS1 where */
/* nnn is the dept # and SYS1 the list ID qualifier. */
/* Input: */
/* LINEFILE file containing department hierarchy */
/* */
/* Example call of this program from the command */
/* line would be: */
/* CALL CRTORGLST */
/* */
/* To compile this program use the CRTCLPGM command. */
/* */
DCL VAR(&OPTION) TYPE(*CHAR) LEN(4)
DCL VAR(&INFORM) TYPE(*CHAR) LEN(4) VALUE('*NO ')
DCLF FILE(QSYS/QAOSDIRF)

/* Display the directory entries to an outfile */
DSPDIR OUTPUT(*OUTFILE) OUTFILE(OFCLIB/DEPTS) +
DETAIL (*FULL) OUTFILFMT(*TYPE2)

/* Call the program CTRLSTS to delete and recreate the*/
/* department distribution list for the organization. */
CALL PGM(OFCLIB/CTRLSTS)

/* Use the DSPDIR outfile. */
OVRDBF FILE(QAOSDIRF) TOFILE(OFCLIB/DEPTS)
```

Figure 3-3 (Part 1 of 2). CRTORGLST CL Program

```

/* Now process each record in the personnel file and */
/* add it to the departments in the organization. */
READF:  RCVF      RCDfmt(OSDIRF)
        MONMSG   MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
        CHGVAR  VAR(&OPTION) VALUE('*ADD')
        CALL    PGM(OFLIB/UPDORGLST) PARM(&DFDEN &DFDGN +
        &DFDPT &OPTION)
/* If the user ID was not found, signal a first */
/* level message to inform the caller of this at the */
/* end of the program. &OPTION is returned from the */
/* UPDORGLST program. */
IF (&OPTION *EQ '*ERR') (CHGVAR VAR(&INFORM) VALUE(*YES))
GOTO READF

EOF:    IF      COND(&INFORM *EQ '*YES') +
        THEN(SNDPGMMSG MSG('Look at the second +
        level text for users that were +
        not added to the department distribution +
        lists. The diagnostic error is CPD8978'))

END:    RETURN  /* Return control to the calling program. */
        ENDPGM

```

Figure 3-3 (Part 2 of 2). CRTORGLST CL Program

The program CRTLSTS that CRTORGLST calls is:

```

PGM
/* Program name: CRTLSTS */
/* This program will create all the organization's */
/* department distribution lists from the LINEFILE */
/* file. The department lists have the name */
/* DEPTnnn SYS1 where nnn is the department number */
/* and SYS1 is the list ID qualifier. */
/* */
/* Input: */
/* LINEFILE file */
/* */
/* Example call of this program from the command */
/* line would be: */
/* CALL CRTLSTS */
/* */
/* To compile this program use the CRTCLPGM command. */
/* */
DCL    VAR(&CATDEPT) TYPE(*CHAR) LEN(7)
DCLF   FILE(OFLIB/LINEFILE) /* File +
        containing organizations departments */

```

Figure 3-4 (Part 1 of 2). CRTLST Program

```

/* Delete and recreate all the department distribution*/
/* lists in the organization. */

READF:      RCVF RCD_FMT(DEPTRCD)
/* Monitor for EOF message */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
CHGVAR      VAR(&CATDEPT) VALUE('DEPT' *CAT &LINDPT)
/* Delete the department distribution list and */
/* monitor for CPF9052 - Distribution list not found */
DLTDSTL     LSTID((&CATDEPT SYS1))
MONMSG      MSGID(CPF9052) EXEC(GOTO CMDLBL(CREATE))
/* Create the new department distribution list */

CREATE:     CRTDSTL     LSTID(&CATDEPT SYS1) LSTD(&CATDEPT)
GOTO READF

EOF:
END:        RETURN      /* Return control to the calling program. */
ENDPGM

```

Figure 3-4 (Part 2 of 2). CRTLST Program

Updating the Directory from Customer Data

The following examples illustrate some of the uses of the directory services commands. They are not full function solutions. They suggest some of the uses you might have for such commands. Also, the commands should give you ideas about how they would fit into your application and computer system.

In the following examples, the creation of user profiles and the addition of these users to the system directory are automatic. When a new employee record is added to the personnel database file, the program shown can be called to automatically create a user profile for the employee and add it to the system distribution directory. With additional coding, the program can make the appropriate changes to the user profile and system directory when an employee record is changed (address, telephone number, and so on) or deleted.

Example 1

In this example, the personnel file is used to create user profiles for each employee in the file. The profiles are then added to the system directory. This example could be used by a company installing a new system or migrating from another system, where the extensive creation of user profiles would be required. The program would be run once in a batch mode.

The ADDDIRE command is used in this example to add a directory entry to the system directory.

Assumptions: It is assumed that the personnel file already exists. In order to make sure each employee has a unique name, the first 6 characters of the last name are concatenated with the first 2 characters of the first name. For example, the user ID for an employee named Tom Connor would be CONNORTO. If many duplicate names occur, additional programming may be required.

The resulting field is used as the user ID for the user profile, the default password, and the user ID portion of the directory entry.

It is also assumed that company policy requires that the profile and the user ID be the same. If they differ, additional programming may be required. The second part of the user ID is an address taken from the department field in the personnel file. If Tom Conner were in the sales department, the address would be SALES. The description field in the system distribution directory entry is built from the last name in the personnel file, followed by the first name. Tom Connor's description would be CONNOR, TOM.

Note: The user address must be the system name when using the VM/MVS bridge.

A default company address is given for each user. The telephone number is updated from the personnel file.

These examples also assume you have the required authority to create user profiles and add system directory entries.

The example personnel file has the following record format:

```
R  PERSRF
   EMP#      8
   FNAME     15
   LNAME     20
   ADDR1     20
   CITY      12
   STATE     2
   ZIP       9
   DEPT      8
   PHONE     15
K  EMP#
```

Figure 3-5. Record Format for the Example Personnel File

Following is the code for the example.

```

PGM /*Program PERS1 reads a personnel file, creates a user
      profile for each employee, and adds the entry to +
      the System Directory */

DCLF      FILE(library/file) /* Personnel file to +
          be used as input to the program */
DCL       VAR(&USERID) TYPE(*CHAR) LEN(8) /* User id +
          field to update directory */
DCL       VAR(&DESC) TYPE(*CHAR) LEN(35) /* Description +
          field to update directory */
START:   RCVF      RCDfmt(filerrf) /* Reads employee record from +
          the record format in the personnel file */
MONMSG   MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF)) /* When +
          the end of file is reached, go to the end of +
          the program */
CHGVAR   VAR(&USERID) VALUE(%SST(&LNAME 1 6) *TCAT +
          (%SST(&FNAME 1 2))) /* Concatenates first +
          6 characters of last name and first 2 +
          characters of first name to form the user +
          id */
CHGVAR   VAR(&DESC) VALUE(&LNAME *TCAT ' ' *CAT +
          &FNAME) /* Build directory user description +
          by concatenating LNAME and FNAME */

```

Figure 3-6 (Part 1 of 2). ADDDIRE command Program

```

CRTUSRPRF USRPRF(&USERID) INLPGM(QOFC/QOFINLPG) /* +
          Enroll the user as a system user by creating +
          a user profile. The ID created in the +
          previous statement is used as the USRID, and +
          the initial program to call for the user is +
          "QOFINLPG." All the other parameters in this +
          command default to the system provided values +
          which means that at first the password will +
          be the same as the USRID. Additional coding +
          would be required if this is not satisfactory +
          for your installation */
MONMSG   MSGID(CPF2214) /* An error will occur if a user +
          profile already exists with that USRID. +
          monitor for this error message. */
ADDDIRE  USRID(&USERID &DEPT) USRD(&DESC) +
          USER(&USERID) ADDR1('Address line 1') +
          ADDR2('Address line 2') +
          TELNBR1(&PHONE) TEXT(&DESC) /* Program adds +
          entry to System Directory */
MONMSG   MSGID(CPD8983) /* Monitor for existing +
          directory entry */
GOTO     CMDLBL(START) /* Go back and read next employee +
          record in personnel file */
EOF:     ENDPGM

```

Figure 3-6 (Part 2 of 2). ADDDIRE command Program

In the example, the personnel file that contains the employee records is declared. Some variables are set up to build the user ID and description fields required by the Create User Profile (CRTUSRPRF) command and the ADDDIRE commands. The first record is read from the personnel file. A MONMSG command determines when the last record is read, and ends the program. If the record is valid, the user

ID field is built by concatenating the first 6 characters of the last name with the first 2 characters of the first name.

I
I

The description field is built by concatenating the last name, a comma, and then the first name. Then, the user profile is created. The user profile is the same as the user ID, and the password automatically becomes the same as the user ID. The initial program to call for the user is OfficeVision/400 (if a user profile already exists with this name, this process is skipped).

It is assumed that each user ID is unique. This program does not verify the uniqueness of each user ID. To do so would require the creation of a routine that compares the new user ID with existing user IDs. If the new user ID already existed, a different ID would be created for the employee.

The ADDDIRE command requires a user ID, user address, and user description. In this example, the created user ID is used for the first part of USRID, and the department number from the personnel file is used for the second part of USRID. For the user description, use the Last name, First name from the personnel file. For detailed information on the restrictions and guidelines for creating IDs, addresses, and descriptions, see the manual *Planning For and Setting Up OfficeVision/400**.

The telephone number is taken from the personnel file as well. With the ADDDIRE command, you can enter additional information such as follows:

- Up to four mailing address lines
- Two telephone numbers
- A location parameter
- Text field (user-defined)
- Whether the user is direct or indirect
- Whether personal mail should be printed

Example 2

This example can be used with the personnel file for continuous maintenance of user profiles and the system directory. As changes are made to the personnel file (a new employee is added, an employee is deleted, or changes are made to current users), the changes should be reflected on the remainder of the system by adding or deleting user profiles, and changing, adding, or deleting entries in the system directory. The program presented here would be only a part of the application (which would consist of other CL or HLL programs).

For example, an HLL program could display the fields for entering the usual personnel information (name, department number, and so on) in addition to whether this is the addition of a new employee, a deletion, or a change. The HLL program could then call the CL program, and pass to it the information and flags required to perform the desired function.

Following are the OS/400 office commands used in this example:

ADDDIRE Add Directory Entry. Adds a directory entry to the system directory.

CHGDIRE Change Directory Entry. Changes data for a user.

RMVDIRE Remove Directory Entry. Deletes an entry from the system directory. When a user profile is deleted, the corresponding entry in the directory is deleted.

Assumptions: In this example, it is assumed that the personnel file already exists (see “Assumptions” on page 3-32 for a description of the file record format). Also, it is assumed that user profiles and the system directory already exist. The information required by this program is being passed by another program responsible for creating the necessary parameters (such as user ID from the last and first name, as described in “Assumptions” on page 3-32), and controlling the flow of information through the application.

Following is the code for this example:

```
PGM PARM(&ADDFLG +
        &CHGFLG +
        &DLTFLG +
        &USRID +
        &ADD +
        &DSC +
        &ADDR1 +
        &ADDR2 +
        &ADDR3 +
        &ADDR4 +
        &TEL1 +
        &TEL2 +
        &LOC +
        &SYSNAM +
        &SYSGRP +
        &INDUSR +
        &PRTPER)
/* Declare variables to receive parameters passed from other program */
DCL     VAR(&ADDFLG) TYPE(*CHAR) LEN(1)
DCL     VAR(&CHGFLG) TYPE(*CHAR) LEN(1)
DCL     VAR(&DLTFLG) TYPE(*CHAR) LEN(1)
DCL     VAR(&USRID) TYPE(*CHAR) LEN(8)
DCL     VAR(&ADD) TYPE(*CHAR) LEN(8)
DCL     VAR(&DSC) TYPE(*CHAR) LEN(50)
DCL     VAR(&ADDR1) TYPE(*CHAR) LEN(36)
DCL     VAR(&ADDR2) TYPE(*CHAR) LEN(36)
DCL     VAR(&ADDR3) TYPE(*CHAR) LEN(36)
DCL     VAR(&ADDR4) TYPE(*CHAR) LEN(36)
DCL     VAR(&TEL1) TYPE(*CHAR) LEN(25)
DCL     VAR(&TEL2) TYPE(*CHAR) LEN(25)
DCL     VAR(&LOC) TYPE(*CHAR) LEN(36)
DCL     VAR(&SYSNAM) TYPE(*CHAR) LEN(8)
DCL     VAR(&SYSGRP) TYPE(*CHAR) LEN(8)
DCL     VAR(&INDUSR) TYPE(*CHAR) LEN(5)
DCL     VAR(&PRTPER) TYPE(*CHAR) LEN(5)
/* Determine which type of maintenance is required */
IF COND(&ADDFLG *EQ 'Y') THEN(GOTO CMDLBL(ADD))
IF COND(&CHGFLG *EQ 'Y') THEN(GOTO CMDLBL(CHG))
IF COND(&DLTFLG *EQ 'Y') THEN(GOTO CMDLBL(DLT))
SNDPGMMSG MSG('**** NONE OF THE FLAGS WERE RECEIVED ****') +
  TOMSGQ(QSYSOPR) MSGTYPE(*INFO)
GOTO CMDLBL(END)
```

Figure 3-7 (Part 1 of 2). Example 2 Program

```

/* Begin the "ADD" routine */
ADD:   CRTUSRPRF  USRPRF(&USRID) INLPGM(QOFC/QOFINLPG) /* Create +
      user profile USRID and initial program to call is AS/400 +
      Office. All other parameters default */
      MONMSG MSGID(CPF2214) /* Monitor for an existing user +
      profile with the same USRID */
      ADDDIRE USRID(&USRID &ADD) USRD(&DSC) USER(&USRID) +
      SYSNAME(&SYSNAM &SYSGRP) INDUSR(&INDUSR) PRTPERS(&PRTPER) +
      ADDR1(&ADDR1) ADDR2(&ADDR2) ADDR3(&ADDR3) ADDR4(&ADDR4) +
      LOC(&LOC) TELNBR1(&TEL1) TELNBR2(&TEL2) /* Add an entry +
      to the System Directory */
      MONMSG MSGID(CPD8983) /* Monitor for an existing directory +
      entry */
      GOTO CMDLBL(END)
/* Begin the "CHANGE" routine */
CHG:   CHGDIRE USRID(&USRID &ADD) USER(&USRID) SYSNAME(&SYSNAM +
      &SYSGRP) INDUSR(&INDUSR) PRTPERS(&PRTPER) ADDR1(&ADDR1) +
      ADDR2(&ADDR2) ADDR3(&ADDR3) ADDR4(&ADDR4) LOC(&LOC) +
      TELNBR(&TEL1) TELNBR(&TEL2)
      MONMSG MSGID(CPF9006) /* Monitor for error message if +
      the entry to be changed does not exist. */
      GOTO CMDLBL(END)
/* Begin the "DELETE" routine */
DLT:   RMVDIRE USRID(&USRID &ADD) USRD(&DSC) /* Remove the entry +
      from the directory. */
      MONMSG MSGID(CPF9006) /* Monitor for error message if +
      the entry to be deleted does not exist. */
      GOTO CMDLBL(END)
END:   ENDPGM

```

Figure 3-7 (Part 2 of 2). Example 2 Program

In the example, this program receives the information from the calling program. The calling program passes employee data and the following flags to indicate:

- A new user addition
- A change to a current user
- A deletion of a user

The program decides which type of maintenance must be done by checking the flags and branching accordingly. The maintenance routines are simple tasks.

Chapter 4. Document Distribution Services

This chapter discusses document distribution services, objects to manage the distribution, the command interfaces to this service, and the limitations of document distribution services support. It also provides examples that use control language (CL) commands to maintain document distributions.

Document Distribution Services Overview

Document distribution services allow office users to send electronic mail to a user or group of users and receive documents from other office users. Electronic mail can consist of documents or messages. The documents may contain any type of information such as final-form documents, revisable-form documents, and PC files. It is the sender's responsibility to ensure the recipient can understand and process the information.

Document distribution services allow users to interchange documents with users on other systems by using **Document Interchange Architecture (DIA)** in conjunction with advanced program-to-program communications (APPC) and SNA distribution services (SNADS). DIA contains the rules and organization for the exchange of information between systems, using either document library services or document distribution services.

In an office network, users can be both senders and recipients of documents. Typically, **sender** refers to a user who sends documents and **recipient** refers to a user who receives documents. Every distribution sent by a sender contains a list of user IDs and addresses for the users receiving the distribution. Each recipient is identified by a two-part user ID/address and a two-part **destination system** name and system group in the system distribution directory. Users can send documents and messages to one or more recipients, who may be local or remote users. Local users receive mail on the same system as you. Remote users receive mail on another system in the network. Distributions for remote users are sent by SNADS or Virtual Machine/Multiple Virtual Storage (VM/MVS) bridge to the remote systems.

When a distribution is sent, one copy is sent to each system and is shared by all recipients on that system.

Document distribution services allow you to send and receive documents and messages on behalf of another user if you have been given authority to do so. Working on behalf of another user gives you the same access to objects and services as the user on whose behalf you are working. For example, you can send and receive mail for another user, but you cannot access distributions marked as personal while working on behalf of another user.

Using Document Distribution Services to Send, Receive, and Manage Mail

Users can write programs with document distribution commands to access the document distribution services by using either batch or interactive jobs. Users calling programs that contain the document distribution commands must be enrolled in the system distribution directory.

The document distribution services commands support sending and receiving documents in the document interchange network. The documents can contain any type of information, such as final-form documents, revisable-form documents, and PC files. It is the sender's responsibility to ensure the receiver can understand and process the information.

Use the document distribution services commands to write programs to perform the following functions:

- Determine whether there is new mail for a user
- Receive distributions from the document interchange network and process information without operator action
- Receive distributions from the document interchange network and forward or reply to the distribution without end user action
- Send output to office users from applications as a part of the applications
- Create batch reports on the use of document distributions
- Perform batch maintenance on document distribution objects

This is not intended to be a complete list, but rather some of the functions that can be performed by CL programs using the document distribution service commands.

Mail Logs

Document distribution services uses the distribution recipient index created for each local user in the system distribution directory to manage three mail logs for each user. The three logs are:

- | | |
|-------------------------|---|
| In-mail log | This log contains new mail that the user has received, old or unopened mail the user has not viewed or handled yet, opened mail that the user has viewed or printed, mail that the user filed in the local or remote document library, or printed mail that the user has logged. Entries stay on this log until deleted. |
| Confirmation log | This log contains the status of mail sent with the confirmation of delivery (COD) option. The status of the distribution is available for each recipient. If there is a distribution error for one or more of the recipients of a distribution that was sent with the COD option, the error is indicated in this log. Entries remain on this log until deleted. |
| Error log | This log contains the error status for mail sent without the COD option, where a distribution error occurred for one or more of the recipients. Entries remain on this log until deleted. |

End-user office products may not show the user these three separate logs. They may be combined into one or more logs.

Relationship to SNADS

Document distribution services uses SNADS to route distributions to recipients at remote nodes. SNADS is a distribution service that distributes objects, such as documents, files, input streams, and messages to other systems directly or indirectly connected to your system. A SNADS configuration must exist on all systems that use the distribution services.

A distribution request can come from a sender who assumes a recipient is located at one receiving system, but that system's distribution directory indicates that the recipient is located at another system. The redirection function of SNADS automatically forwards the distribution to the system indicated in the distribution directory for that recipient.

The redirection function allows users to move from one system to another without changing all the system distribution directories on all the systems in the SNADS network. The system distribution directory where the user was a local user and the system distribution directory to which the recipient has moved are the only directories that require change. When any distribution is routed to the recipient's former location, the system name/system group in the distribution is updated with the value in the system distribution directory, and the distribution is forwarded to the recipient's current location. It is recommended that you update the directories in all network nodes as soon as possible to make the most efficient use of your network.

VM/MVS Bridge

The VM/MVS bridge is an application that provides distribution services between an AS/400 SNADS network and a VM/370 Remote Spooling Communications Subsystem (RSCS) network. Distributions created by document interchange can be sent and received over this bridge. Object distribution can also send and receive using this bridge. Object distribution is a control program application that works under SNADS and distributes objects other than documents (data files including save files, spooled files, input streams, and messages). The VM/MVS bridge allows users of OfficeVision/400, DISOSS, or any DIA/SNADS node connected to the bridge to exchange final-form documents, revisable-form documents, notes, messages, and PC files with the OfficeVision/VM users. Users can send documents, notes, and messages with or without memo slips.

The end user does not see the VM/MVS bridge. Instead, it is transparent and makes it appear that the user is communicating with another user on the same type of network. For more information, refer to the *Distribution Services Network Guide*.

OSI Message Services/400

OSI Message Services/400 is a licensed program that can route data in an open systems interconnection (OSI) network between two or more network nodes. OSI Message Services/400 acts as a distribution service that can store data for delayed delivery and can distribute documents, files, and AS/400 messages. In X.400 terminology, all of these AS/400 objects are called messages.

In OSI Message Services/400, a user agent is used to create, send, and receive the AS/400 notes and messages. A user agent can be either OfficeVision/400 or the Document Interchange Architecture (DIA). The AS/400 object distribution commands (for example, the Send Network Message (SNDNETMSG) and Send Network Spool File (SNDNETSPLF) commands) can be used to send and receive files, spooled files, and messages across an OSI network.

When OSI Message Services/400, SNADS, and TCP/IP are configured, users do not need to know the type of network on which remote users are located. In each case, the local system handles the proper routing.

OSI Communications Subsystem/400 is required before OSI Message Services/400 can be used.

Simple Mail Transfer Protocol

The Simple Mail Transfer Protocol (SMTP) is used to send or receive electronic mail. For consistency with other AS/400 mail functions, SMTP uses SNADS, which contains extensions to support SMTP. Because SMTP uses SNADS, users can send mail to various types of users (not just SMTP users) with one consistent user interface. The distribution services (receive, forward, and send electronic mail) for OfficeVision/400 functions are provided by SNADS.

SMTP supports the following functions:

- Both client and server functions on the AS/400 system
- OfficeVision/400 notes and messages
- AS/400 documents in final-form text (FFT) format
- Documents and messages through SNADS using IBM Displaywriter or Personal Services/PC
- Documents and messages using AS/400 commands

Note: If OfficeVision/400 is not installed, users can still send and receive SMTP messages.

- SMTP notes as an intermediate TCP/IP step on an SMTP distribution
- Binary file transport across TCP/IP links

SMTP does not support the following:

- Distributing objects
- Transferring revisable-form text (RFT) documents

Note: If a document is saved on the AS/400 system in RFT format, the document format must be changed to FFT before sending.

- Sending information using the Send Network Message (SNDNETMSG), Send Network Spooled File (SNDNETSPLF), or Send Network File (SNDNETF) command
- Transferring documents exceeding 16 million bytes

Document Distribution Services Objects

The following objects are used by document distribution services:

- Distribution Recipient Index (*DRX)
- Distribution Tracking Object (*DTO)
- Document Unit Object (*DUO)

Figure 4-1 shows the objects used by distribution services.

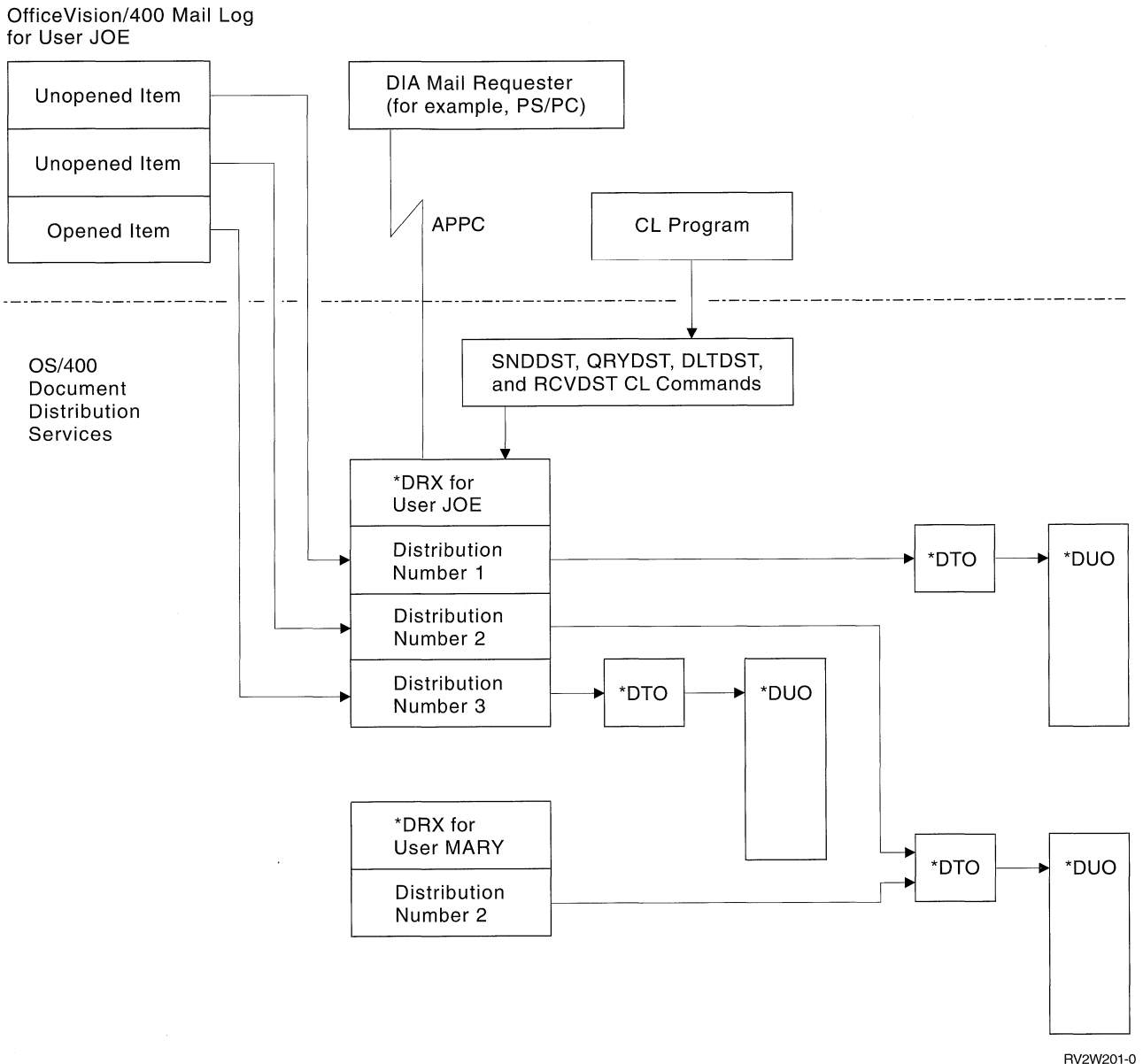


Figure 4-1. Distribution Objects

These objects can be managed using the following:

- The mail functions of OfficeVision/400
- Control language (CL) commands
- Other office products that use DIA distribution services, such as Displaywriter

Objects created and managed by document distribution services are internal objects and are only accessible by using document distribution services. These objects are owned by the QDOC profile. Storage used is charged to the QDOC profile. The QDOC profile is shipped with the system and has no maximum storage limit. The objects can be accessed for debugging by using the Dump System Object (DMPSYSOBJ) command.

Distributions can only be accessed by the intended recipients and cannot be shared by users other than those authorized to work on behalf of the recipients. Therefore, distribution objects are not visible and cannot be accessed by OS/400 utilities and commands, such as the Work with Library (WRKLIB) or the Move Object (MOV OBJ) commands.

Distribution Recipient Index

When a local user is enrolled in the system distribution directory, a **distribution recipient index (*DRX)** is created in library QUSRSYS. The *DRX contains an entry for each incoming distribution and for each incoming document distribution. The *DRX also contains entries for outgoing distributions when the sending user requests confirmation of delivery or when an error occurs during the distribution. These entries stay on the *DRX until deleted by either the end user or through the CL commands (DLTDST) or RCVDST KEEP(*NO) for incoming distribution. If library QUSRSYS is lost, cleared, or deleted, the *DRXs are lost. The *DRXs can be restored by using the save/restore function for mail (see "Individual Object Restore" on page 9-3). If the system directory files are restored, the *DRXs are re-created when needed. The *DRX is deleted when a user is removed from the system distribution directory.

Distribution Tracking Object

Each office distribution is controlled by a **distribution tracking object (*DTO)**. The entries on the sender and recipient *DRXs point to the *DTO for the distribution. If library QUSRSYS is lost, cleared, or deleted, then the *DTOs are lost, and the distributions that they controlled are lost as well. These *DTOs can be restored by using the save/restore functions for mail.

A *DTO object is deleted when all the recipients and the sender have deleted the distribution from their mail logs.

Document Unit Object

Each office distribution of a document also has a **document unit object (*DUO)**. The *DTO for the document distribution points to the *DUO for the distribution. This object contains the document content and the document details.

If library QUSRSYS is lost, cleared, or deleted, the *DUOs are lost. These can be restored using the save/restore functions for mail. A *DUO object is deleted from the system when the *DTO is deleted.

Dumping Internal Mail Logs

All *DRX, *DTO, and *DUO objects are internal objects that can be accessed only by using document distribution services. For problem analysis, these objects can also be accessed using the Dump System Object (DMPYSOBY) command.

To dump a specific user's *DRX, type the following:

```
DMPYSOBJ OBJ('UserIDAddress') CONTEXT(QUSRSYS)
TYPE(0E) SUBTYPE(D1)
```

This command produces the output shown in Figure 4-2:

```
OBJECT TYPE-          INDEX
NAME-      ADAFIN  RCHAS367          TYPE-      0E  SUBTYPE-      D1
LIBRARY-   QUSRSYS                                TYPE-      04  SUBTYPE-      01
CREATION-  10/04/90  6:15:27          SIZE-      00002600
OWNER-     QDOC                                TYPE-      08  SUBTYPE-      01
ATTRIBUTES- 0800                                ADDRESS-    00079E00  0000
INDEX ATTRIBUTES-
000000  00000000  00000071  0ED1C1C4  C1C6C9D5  4040D9C3  C8C1E2F3  F6F74040  40404040  *      É  JADAFIN  RCHAS367  *
000020  40404040  40404040  E0000000  00000000  000005E0  00000000  00000000  00000000  *      \      \      *
000040  00000000  00000000  00010002  7E000400  00000000  00000000  00000000  00000000  *      =      *
000060  30007800  19000000  00000000  0B000000  0E                                *      Ì      *
INDEX ENTRIES-
. 00001-
000000  D4D9D9C3  C8C1E2F3  F6F7C1C4  C1C6C9D5  4040F0F0  F0F2F0F1  01000000  00000000  *MRRCHAS367ADAFIN  000201  *
000020  00000000  00000000  003B0123  A6001900  00000000  00000000  00000000  00000000  *      w      *
000040  00000200  01D9C3C8  C1E2F3F6  F7C1C4C1  C6C9D540  40F0F0F0  F2404040  40404040  *      RCHAS367ADAFIN  0002  *
000060  40404007  C60A050F  27000E07  C60A050F  281A3200  00000000  *      F      F      *
..POINTERS-
000020  SYP  19  E2  RCHAS367ADAFIN  0002          04  01  QUSRSYS          0000  *DTO
. 00002-
000000  D4D9D9C3  C8C1E2F3  F6F7C1C4  C1C6C9D5  4040F0F0  F0F2F0F1  02000000  00000000  *MRRCHAS367ADAFIN  000201  *
000020  00020000  0000A388  89A24089  A240A388  85409596  A38540A2  A4829185  83A34040  *      this is the note subject *
000040  40404040  40404040  40404040  40404040  40400000  00000000  000007C6  0A050F27  *      F      *
000060  0C0F0000  00000000  00000000  00000000  00000000  00000000  *
..POINTERS-
NONE
ASSOCIATED SPACE-
000000  00000003  20100000  000180D4  00000000  80D60000  00010000  00000000  00000000  *      M  0      *
000020  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  *
LINES 000040 TO 0000BF SAME AS ABOVE
0000C0  00000000  00000000  00000001  00000000  00000000  00000000  00000000  00000000  *
0000E0  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  *
LINES 000100 TO 0005DF SAME AS ABOVE
..POINTERS-
NONE
OIR DATA-
NONE
END OF DUMP
***** END OF LISTING *****
```

Figure 4-2. Dumping a Distribution Recipient Index

To dump the specific *DTO referred to by the message on the *DRX, type the following:

```
DMPYSOBJ OBJ('RCHAS367ADAFIN 0002') CONTEXT(QUSRSYS)
TYPE(19) SUBTYPE(E2)
```

To dump a specific *DTO on the system (received from a remote user), type the following:

```
DMPYSOBJ OBJ('Q0SRDIST@yymddhhmssmsc')
CONTEXT(QUSRSYS) TYPE(19) SUBTYPE(E2)
```

In the previous example, 0yymddhhmssmsc equals the date and time when distribution was received at this system. For example, 0890615103603057 is for 06/15/89 10:36:03:57. It is unlikely that you will know the exact time you can type Q0SRDIST@yymdh* and dump all the *DTOs for the hour you enter.

To dump all the *DUOs on the system, type the following:

```
DMPYSOBJ OBJ(*ALL) CONTEXT(QUSRSYS)
TYPE(19) SUBTYPE(E3)
```

Document Distribution Services Application Interfaces

You can access document distribution services functions for sending and receiving documents and messages by using an end-user office product like OfficeVision/400 or by using the document distribution commands. The OfficeVision/400 licensed program provides an interactive menu/display interface for sending and receiving electronic mail. You will also need an interface to use in batch applications. This interface allows you to write your own menu/display driven application. You can write programs containing these commands to access the document distribution services by using either batch or interactive jobs. You must be enrolled in the system distribution directory to call programs containing the document distribution command.

The document distribution services commands support the following kinds of user-written functions (this is not intended to be a complete list):

- Send a message, a document, or a data file to a user or group of users. The data file could be a database file or it could be a file containing document content architecture data streams (FFTDCA, RFTDCA, and so on) built by a customer application.
- Query the distribution recipient index to determine:
 - Whether there is any incoming mail
 - Status of outgoing mail
 - Whether an error occurred
- Receive distributions from the document interchange network and process information without operator action.
- Receive distributions from the document interchange network and forward or reply to the distribution without end user action.
- Send output to office users from applications as a part of the applications.
- Create batch reports on the use of document distribution services.
- Perform batch maintenance on document distribution services objects.
- Delete an incoming distribution or the status for an outgoing distribution from a recipient index.
- Receive a distribution and place the data into a database file.
- Receive a distribution and file the data into the document library.

A user has a single recipient index, and it is used by the system to support the command, the OfficeVision/400 interface, and the document interchange interface. Any application that uses the command can be used with these other products. For example, a program can be used to send a note. The note can be viewed, printed, and so on by an OfficeVision/400 user.

Other Document Distribution Services Topics

This section discusses damaged objects, jobs in the QSNADS subsystem, restrictions and messages, and security considerations.

Damaged Objects

There are several ways a *DRX can become damaged, including:

- A machine failure
- A disk unit error
- A program problem
- A power failure when a distribution is sent, received, changed, or canceled

After one of these problems occurs, start the system to automatically recover most of the damaged *DRXs. When a damaged *DRX is accessed while the system is active, the system deletes the index and rebuilds it again in the current job.

Another way to recover damaged *DRXs is to call the IPL *DRX recovery function directly CALL PGM(QSYS/QOHFIXIX) PARM(Y). Only the controlling subsystem can be active, and the CALL must be the only job in the controlling subsystem. The *DRX recovery program rebuilds the *DRXs from information in the *DTOs.

A *DTO or *DUO can be damaged by a disk unit error or program problem. When a damaged *DTO or *DUO is found, it is deleted and the status of the mail log entry is set to DAMAGED on the *DRX. To recover the mail log item, restore all of that user's mail or request that the sender resend the mail.

Document Distribution Services Jobs in the QSNADS Subsystem

The processes controlling the functions of the SNADS network run in the QSNADS subsystem. IBM-supplied transaction programs, such as document interchange and object distribution, also run in the QSNADS subsystem.

You must start the QSNADS subsystem before you use OfficeVision/400 to perform tasks such as filing or sending documents regardless of whether the AS/400 system is part of a SNADS network. Refer to the Start Subsystem (STRSBS) command in the *CL Reference* manual for more information.

Programs such as document interchange (QDIA) and object distribution (QNFTP) handle distributions originated by remote users and delivered to local users. For OfficeVision/400 users, QDIALOCAL, the internal job that runs in the QSNADS subsystem, handles all file document and send requests for OfficeVision/400.

The following DIA jobs run in the QSNADS subsystem. These jobs run under an IBM-supplied job description called QDIA.

QDIA This job is the DIA destination application transaction program. It receives distributions that originated on a remote system from SNADS, and performs the local distributions. This job waits for distributions to be received. If none are received, the job takes no system resources.

This job runs under the QSNADS class.

QDIALLOCAL This DIA job processes asynchronous send distribution and file document requests. Asynchronous distribution requests exist when a user sends, forwards, or replies to a note through OfficeVision/400, or when a user works with documents in folders using the WRKDOC command and does a send request. Asynchronous file requests exist when a user files a piece of mail from the mail log or files the piece of mail during the send request. This job waits for requests. If none are received, this job takes no system resources.

This job runs under the QDIALLOCAL class. You can change the run priority for this job by using the Change Class (CHGCLS) command.

QDIINDUSR This DIA job handles the indirect user mail support function. Mail can be distributed to a user who is defined in the system distribution directory as an indirect user (see Chapter 3, "Directory Services," for information about indirect users). If there are no indirect users defined in the directory, this job waits until an indirect user is defined. While the job waits, it takes no system resources.

This job runs under the QINDUSR class. The default wait time for this job is 5 minutes. If there is no mail for any indirect users, the job stops itself for 5 minutes. It then starts again and checks to see if any new mail has arrived for any indirect users. This process is repeated as long as the job is running in the QSNADS subsystem.

If the default wait time of 5 minutes is not acceptable, you can do one of two things:

- Use the Change Class (CHGCLS) command to specify your choice of a default wait time.
- Use the Change Job (CHGJOB) command to change the default wait time. The change is only in effect during the time the job is running in the QSNADS subsystem. If the subsystem is ended at any time and started again, the CHGJOB command must be issued again to change the default wait time.

QDIANOTIFY This job is no longer being submitted in the QSNADS subsystem.

This job runs under the QMAILP class.

For help in problem analysis, you may display any of these jobs by using the Display Job (DSPJOB) command or the Display Job Log (DSPJOBLOG) command.

Restrictions/Limitations

This section describes the limits and restrictions of document distribution services support.

Document distribution services provide full distribution support for all documents. OfficeVision/400 supports final-form text, revisable-form text, and System/38 documents. Refer to the *Final-Form Text Reference* and the *Revisable-Form Text Reference* manuals for details. All documents sent to an office user, regardless of

document type, are listed in the user's mail log. An office user can send and receive any document type. If the type is not final-form, revisable-form, or System/38 text, the user cannot view, print, or edit the document, but it can be filed, sent to another user, or processed using the document distribution services CL commands.

Host nodes support image programmable work stations (DISOSS/370 provides document distribution services to a Scanmaster). You can send final-form text documents from any AS/400 work station to a user on a Scanmaster attached to DISOSS/370, but the Scanmaster cannot send documents to you unless your work station supports image data streams. Scanmaster can receive either final-form text or image data streams, but can send only image data streams.

OfficeVision/VM can be attached through DISOSS to the AS/400 system. When connecting directly from OfficeVision/VM to the AS/400 system, you must exchange documents through the VM/MVS bridge support. (See the *Distribution Services Network Guide* for more information about this support.)

Distributions being processed by document distribution services can be lost after a device error or loss of power. For each distribution, internal objects contain and track the distribution. If all the objects are lost, neither the sender nor recipient is notified. If only one of these internal objects is lost, the sender is notified of the failure. New mail should be received by the recipient and filed in the document library if the user receiving the documents wants to avoid unexpected loss of information.

New Mail Notification

A system message is sent from document distribution services to each user (direct or indirect) for the receipt of an office distribution in the following situations:

- Priority mail, personal mail, and messages
- Priority and personal mail
- All office mail

Based on what is established in the system distribution directory for each user, you can specify whether you want to be notified of the arrival of mail or the arrival of certain types of mail (priority and personal mail or messages) through the user's message queue. You can indicate notification for specific types of mail, notification for all mail, or no notification. If you choose specific types of mail, then you can choose from the following:

- Priority and personal mail
- Messages

The system user profile is used to determine whether a local recipient has a system message queue, the name and library of the queue, and how the messages are to be delivered. If a local user does not have a system message queue (no system user profile message queue) or the user's system message queue is full, the messages are not sent to that recipient.

If the distribution is not personal, the message is sent to the address. The recipient's message queue contains the distribution description and the office message. If the distribution is personal, no user information is included.

Messages are translated from the system or display station character identifier (CHRID) to a common character identifier for distribution. When the messages are sent to the recipient's message queue, they are translated to the system character identifier.

Messages sent by document distribution services to user system message queues can be received by the RCVMSG command. The message IDs are listed in Figure 4-3.

Figure 4-3. Message IDs

| Message ID | Severity | Conditions |
|------------|----------|---|
| CPI9095 | 50 | Personal, nonpriority distribution |
| CPI9096 | 80 | Personal, priority distribution |
| CPI9097 | 50 | Nonpersonal, nonpriority message only |
| CPI9098 | 80 | Nonpersonal, priority message only |
| CPI9099 | 80 | Nonpersonal, priority distribution |
| CPI904B | 50 | Nonpersonal, nonpriority documents or notes |

Printing Mail With or Without a Cover Page

If you wish to print mail, you can specify in the system distribution directory whether a cover page is to be printed along with the mail. If a message is printed, the cover page is automatically printed, because the message resides on the cover page. If a note is printed that has a memo attached, and the print cover page indicator is set to NO, the cover page is not printed.

Indirect User Data Queue Interface

An interface is provided to allow you to write an application to generate your own indirect user function. Information for all indirect users is written to a user-defined data queue. For each indirect user who is on a recipient or distribution list, one entry of information is written to the user-defined data queue. The information is retrieved from the user-defined data queue by calling the Receive Data Queue (QRCVDTAQ) program from a high-level language program, RPG, for example. Each call to QRCVDTAQ receives one entry of information from the user-defined data queue. For more information on QRCVDTAQ, see the *CL Reference* manual.

Activate Indirect User Data Queue Function

To activate this function, an IBM-defined data area must exist. The required data area, called QMAILDQ, must be created properly with the correct authority and ownership in order to get the indirect user's information written to user-defined high priority and normal priority data queues. These two data queues must be created with the proper authority and ownership to the object and library.

High and Normal Priority Data Queues: The high and normal priority data queues must be created with a user-defined name and library of your choice. The data queue name and library name can be 1 to 10 characters in length. The maximum length of each data queue entry is 160 characters.

To create the data queues, enter the following command. Use this command twice, once to create the high priority data queue and once to create the normal priority data queue:

```
CRTDTAQ DTAQ(LIBRARY NAME/DATA QUEUE NAME) MAXLEN(160)
      SEQ(*FIFO) AUT(*CHANGE)
```

Note: AUT(*CHANGE) is the default for USER(*PUBLIC). Also, AUT(*CHANGE) allows the sender to have proper authority to write information to the data queue.

To ensure proper ownership of the data queues, enter the following command twice, once for each data queue:

```
CHGOBJOWN OBJ(LIBRARY NAME/DATA QUEUE NAME) OBJTYPE(*DTAQ) NEWOWN(QSYS)
```

Note: QSYS must be the owner of both the high and normal priority data queues.

You must create the data area QMAILDQ in QUSRSYS. This data area is used to determine whether information is to be written to the user defined high or normal priority data queue. Use the following command to create the IBM-defined data area:

```
CRTDTAARA DTAARA(QUSRSYS/QMAILDQ) TYPE(*CHAR) LEN(64)
      VALUE('HIGHPRIOR MAILLIB NORMPRIOR MAILLIB')
      TEXT('MAIL INDIRECT USER DATA AREA')
      AUT(*CHANGE)
```

Note: AUT(*CHANGE) is the default for USER(*PUBLIC) when the QMAILDQ data area is created.

The layout of the VALUE parameter on the CRTDTAARA command is:

| | | |
|------------------|---|---|
| POSITION 1 - 10 | = | High Priority Data Queue Name |
| POSITION 11 | = | BLANK SPACE |
| POSITION 12 - 21 | = | High Priority Data Queue Library Name |
| POSITION 22 | = | BLANK SPACE |
| POSITION 23 - 32 | = | Normal Priority Data Queue Name |
| POSITION 33 | = | BLANK SPACE |
| POSITION 34 - 43 | = | Normal Priority Data Queue Library Name |
| POSITION 44 - 64 | = | NOT USED |

Note: HIGHPRIOR, NORMPRIOR, and MAILLIB are only used as examples in the VALUE parameter on the CRTDTAARA command.

QSYS must be the owner of the data area. Otherwise, the indirect user's information is not written to the data queue. Use the following command to assure QSYS to be the owner of the data area:

```
CHGOBJOWN OBJ(QUSRSYS/QMAILDQ) OBJTYPE(*DTAARA) NEWOWN(QSYS)
```

Note: If you do not run the CRTDTAQ, CHGOBJOWN, CRTDTAARA, and CHGOBJOWN commands, indirect user information is not written to the user-defined data queue.

Security Considerations

If the authority assigned to the data area and data queue is greater than *PUBLIC, use the RVKOBJAUT command to change it. The authority of these objects can be changed from *PUBLIC to *USE or *EXCLUDE.

If the authority level is changed to *USE or *EXCLUDE, then the QDOC and QSECOFR user profiles must be granted *ALL authority to these objects. Enter the following commands to grant the QDOC and QSECOFR user profiles the proper authority:

```
GRTOBJAUT OBJ(LIBRARY NAME/ DATA QUEUE NAME)
           OBJTYPE(*DTAQ)
           USER(QDOC QSECOFR)
           AUT(*ALL)
```

```
GRTOBJAUT OBJ(QUSRSYS/QMAILDQ)
           OBJTYPE(*DTAARA)
           USER(QDOC QSECOFR)
           AUT(*ALL)
```

Other useful commands for the data area include:

To change the data area value, type:

```
CHGDTAARA DTAARA(QUSRSYS/QMAILDQ)
```

To see whether the data area exists and its current value, type:

```
DSPDTAARA DTAARA(QUSRSYS/QMAILDQ)
```

Data Queue Entry Information

Information is written to the user-defined data queue for each indirect user. When mail is sent to an indirect user as high priority, the information is written to the high priority data queue. Otherwise, the information is written to the normal priority data queue. When an error occurs, a message is sent to QSYSOPR message queue. Also, processing for the send request is continued. This means that you still get the mail that was sent to you, but the information may or may not be written to the user-defined data queue.

The data queue information layout for each indirect user follows. The fields are listed in the order in which they occur for each entry in the data queue.

| | |
|--------------------------|------------|
| DISTRIBUTION ID | CHAR(20) |
| RECIPIENT USER ID | CHAR(8) |
| RECIPIENT ADDRESS | CHAR(8) |
| SENDER USER ID | CHAR(8) |
| SENDER ADDRESS | CHAR(8) |
| SENT DATE | CHAR(8) |
| SENT TIME | CHAR(6) |
| PERSONAL FLAG | CHAR(1) |
| PRIORITY FLAG | CHAR(1) |
| CONFIRMATION | CHAR(1) |
| DOCUMENT TYPE | ZONED(5,0) |
| SYSTEM CODE | CHAR(13) |
| DISTRIBUTION DESCRIPTION | ZONED(3,0) |
| CHARACTER SET | |
| DISTRIBUTION DESCRIPTION | ZONED(3,0) |
| CODE PAGE | |

| | |
|---------------------------|------------|
| DISTRIBUTION DESCRIPTION | CHAR(44) |
| DISTRIBUTION ID EXTENSION | ZONED(2,0) |
| MAIL LOG STATUS | CHAR(1) |
| LAST CHANGED DATE | CHAR(8) |
| LAST CHANGED TIME | CHAR(6) |
| UNUSED | CHAR(6) |

This information is the same information that is written to an output file when the QRYDST command is run. For more information on the output file layout for QAOSILIN, see Appendix C, "Office Services Output File Considerations."

Deactivate QDIAINDUSR Job: The QDIAINDUSR job should be deactivated to prevent duplication of function. If the QDIAINDUSR job is active and the user application is running, these jobs may try to process the same piece of mail at the same time. This is a waste of system resources.

A CL program is needed to deactivate the QDIAINDUSR job. Create a source member called INACTIVE. The code needed for the INACTIVE CL program is:

```

/*****/
PGM
ENDPGM
/*****/

```

To create the INACTIVE CL program, enter this command:

```

CRTCLPGM PGM(LIB/INACTIVE)
          SCRFILE(LIBNAME/SOURCE FILE NAME)
          SCRMBR(INACTIVE)
          TEXT('CL PROGRAM TO DEACTIVATE QDIAINDUSR JOB')

```

While QSNADS SUBSYSTEM is not active, enter the Change Routing Entry (CHGRTGE) command. This command changes the routing entry for the QDIAINDUSR job. The routing entry is changed to call the CL program called INACTIVE instead of the QOSMLFMT program. Type this command:

```

CHGRTGE SBSD(LIBRARY NAME/QSNADS)
          SEQNBR(660)
          PGM(LIBRARY NAME/INACTIVE)

```

When the QSNADS subsystem is started, job QDIAINDUSR is submitted to the QSNADS subsystem, but this job ends immediately.

Note: To reactivate the QDIAINDUSR job, another CHGRTGE command must be used. Type this command:

```

CHGRTGE SBSD(LIBRARY NAME/QSNADS)
          SEQNBR(660)
          PGM(QSYS/QOSMLFMT)

```

There is no restriction on creating duplicate QSNADS subsystem descriptions in multiple libraries in the system library list. Make sure the routing entry being changed is for the QSNADS subsystem description used when the STRSBS command is used.

Security Considerations (Working on Behalf of Another User)

Permission to work on behalf of another user can be granted or removed by using the OfficeVision/400 product or by using CL commands. User permission can be given using the Grant User Permission (GRTUSRPMN) command. User permission can be removed using the Revoke User Permission (RVKUSRPMN) command. Users do not need special authority to control which users can work on their behalf. The security officer or an administrator (a user with *SECADM special authority) can use these commands to give or remove permission for any user to work on behalf of any other user.

A mail log can be accessed only by the owner and a user who has been granted permission. The QSECOFR profile or a user with *SECADM authority cannot access another user's mail log. The Grant User Permission (GRTUSRPMN) command must be used to authorize the QSECOFR or *SECADM users to work on behalf of the owner of the mail log.

See "Example Applications" on page 4-29 for instructions on granting user permission.

Document Distribution Services Control Language Interfaces

Refer to the *CL Reference* manual for detailed information about command parameters and error messages created from the CL commands discussed in this section.

A user has a single distribution recipient index used by the system to support the command, the OfficeVision/400 interface, and the document interchange interface. An application that uses the command can be used with these other products. For example, a program can send a document, and this document can be viewed, printed, and so on by an OfficeVision/400 user. The document must be a type that can be processed by an AS/400 system.

To access another user's distributions, you need permission to work on behalf of the other user. You may want to set up your security officer and administrators with this permission, since they are not granted it automatically. To get access to all of the users' distributions, you can write a CL program that lists all the system distribution directory entries to an output file, (DSPDIR), and then use the Grant User Permission (GRTUSRPMN) command with the user profile field from the DSPDIR output file to grant the security officer and administrator authority to work for other users. See "Example Applications" on page 4-29 for an example of how to do this.

Incoming distributions are those distributions that have been received. Outgoing distributions are those distributions that you sent.

The following commands are described in this section:

- Change Distribution Description (CHGDSTD)
- Delete Distribution (DLTDST)
- Query Distribution (QRYDST)
- Receive Distribution (RCVDST)
- Send Document (SNDDOC)
- Send Distribution (SNDDST)

Change Distribution Description (CHGDSTD) Command

The Change Distribution Description (CHGDSTD) command allows a user to change descriptive information such as author, subject, keywords, or access information about a distribution document that is in the mail log. In addition, descriptive information can be removed or added for some parameters.

Security Considerations

A user may change the document description for their own distribution or on behalf of someone else if properly authorized to do so (through the GRTUSRPMN command).

Some Ways to Use This Command

Use the CHGDSTD command to change the details of a distribution.

You can change all of your distributions to personal so no one else can see them. Use the QRYDST command to list your distributions, process each record, and use the CHGDSTD PERSONAL(*YES) command to change the distribution to personal.

When forwarding mail, you may want to change one of the fields, such as KEYWORD, to indicate that it has been forwarded.

Rules

When an existing distribution document interchange document profile parameter is changed and that parameter can have multiple occurrences, the old value must be supplied exactly as it exists in the distribution. For example, if one of the authors supplied with the distribution was John Q. Public, the AUTHOR parameter on this command must contain the exact character string if it is to be changed. When an interchange document profile parameter occurs just once, only the new value must be supplied to change the old value or add the parameter.

You must be enrolled in the system distribution directory to use this command.

Character Set and Code Page Considerations

The following parameters are translated from the value specified in the CMDCHRID parameter to character set 930, code page 500:

- USRID
- DSTID

The following parameters are translated from the value specified in the CMDCHRID parameter to character set 697, code page 500 for the search index database file:

- COMMENT
- DOCD
- AUTHOR
- DOCCLS
- KWD
- SUBJECT
- FILCAB
- CPYLIST

Delete Distribution (DLTDST) Command

The Delete Distribution (DLTDST) command allows a user to delete distributions. A user who signs on and has *ALLOBJ authority can delete damaged and dangling distribution objects. This command also deletes the incoming, outgoing, and error distributions.

The distribution is deleted from the user's mail log but not from the system until the last reference to that distribution is deleted. However, it will no longer be associated with the user's mail log from which it was deleted.

Security Considerations

The requester can delete a distribution and the distribution of another user, if permission to work on behalf of that user was given with the GRTUSRPMN command.

Some Ways to Use This Command

The following are some ways to use the DLTDST command:

- You can write an application that checks to see if a person's outgoing or incoming queue is too large (you determine what is too large). Use the QRYDST command and look at the output file to determine the number of distributions a user has.

If the user has too many distributions, you can send a message warning that the queue is too large and instruct the user to delete some of the distributions. If after some time the queue is still too large, you can do a QRYDST to an output file, use the data from the output file, and automatically delete the oldest distributions using the DLTDST command.

- You can instruct users to keep distributions that are only a certain age, for example, 2 months old. After 2 months, they are automatically deleted. This also keeps the user's queues cleaned up. Your program can do a query to an output file (QRYDST). Then use the Send date field to determine whether the distribution should be deleted, and if so, use the DLTDST command.
- See "Maintaining Mail Logs" on page 4-73 for an example of how to maintain mail logs using the DLTDST command.

Rules

The following rules apply to the DLTDST command:

- You cannot delete another user's mail, even when you are working on behalf of that user.
- Up to 50 distribution IDs can be specified at one time.
- The distribution ID is always 20 characters in length. The first 8 characters are the sender's address padded with blanks, the second 8 characters are the sender's user ID padded with blanks, and the last 4 characters are a 4-digit, zoned-sequence number with leading zeros. When there are blanks, you need to separate the distribution IDs with quotation marks, otherwise quotation marks are not necessary.
- The distribution ID extension takes the default values of 01 for incoming distributions and 00 for outgoing distributions.
- If you do not find one of the distribution IDs in the list, an error is returned and an attempt is made to delete the remaining distributions in the list.

Hints and Techniques

To get the distribution identifier, you can do a Query Distribution (QRYDST) command to an output file and then use the Display Physical File Member (DSPPFM) command to display the output file. The output file has the distribution identifier for each distribution. Also, when a distribution is sent, the distribution identifier is the second level part of the completion message (CPC9033).

Character Set and Code Page Considerations

DSTID and USRID are translated from the value specified in the CMDCHRID parameter to character set 930, code page 500.

Lowercase characters can be typed for the distribution ID and for the user ID because they are translated automatically to uppercase.

Query Distribution (QRYDST) Command

The Query Distribution (QRYDST) command allows a user to list information on incoming or outgoing distributions for a requester or on behalf of another user.

Security Considerations

If the user of this command requests distribution information on behalf of another user, they must have the authority to work on behalf of the other user. This authority may be obtained by using the GRTUSRPMN command.

Security officers and security administrators also need permission to work on behalf of a user to access that user's distributions.

A user working on behalf of another user on a personal mail item does not have access to the document description, sender ID, or address. These are blank.

Some Ways to Use This Command

Following are some ways to use the QRYDST command:

- If you want the status for the completed outgoing distributions deleted automatically, you can use the option DLTSTS(*YES) on the QRYDST command. OfficeVision/400 maintains a separate file when it distributes something. The DLTSTS(*YES) option deletes those entries.
- To find out how many distributions a user has, a count is returned in a completion message (CPC9030). If you specify OUTFILE(*NONE), the QRYDST command only returns a count of NEW and OLD distributions. If you specify an output file on the QRYDST command, a count of all distributions (for example, OPENED, FILED, NEW, OLD, and so on) is returned. You can also query incoming distributions by status. If ALLAUT is specified for the USRID parameter, all distributions the requester is authorized to are listed, regardless of whether the output goes to an output file.
- The QRYDST command puts records of information about distributions to an output file if requested by the OUTFILE parameter. This output file can be used in programs to do the following:
 - You can use QRYDST to create the output file and look at the number of distributions for a user. If the user has too many, you can delete them automatically.

- An application program can be written to receive the distributions. Use the RCVDST command and the distribution ID from the output file created by the QRYDST command.
- If a person is absent for a time, you can write a delegate function to delegate their nonpersonal mail to someone else. Do a QRYDST of the user's incoming mail to an output file and then send the distribution again (SNDDST) to the person delegated to take care of the mail. See “Delegating Mail” on page 4-40 for an example of a CL program that does this function.

Rules

The following rules apply to this command:

- Authority is needed to work on behalf of a user to query the user's distributions. Use the GRTUSRPMN command to do this.
- If the output file does not exist and you specified an output file on the command, it is created for you with the system format.

Output File Layout

There are two types of output files for QRYDST:

- For incoming distributions, the model output file supplied with the system is QAOSILIN in library QSYS, record format OSLIN.
- For outgoing distributions, the model output file supplied with the system is QAOSILOT in library QSYS, record format OSLOUT.

For more information about the layout of the output file for this command, refer to Appendix C, “Office Services Output File Considerations.”

Hints and Techniques

Following are some hints and techniques for using the QRYDST command:

- Use QRYDST to see all of your distributions or another user's distribution (when you work on behalf of that user). Use DSPPFM to display the output file resulting from the QRYDST command.
- When a QRYDST command is used with DLTSTS(*YES) to create an output file, the completed error distributions on the undelivered log are deleted, but show up as entries in the output file. You may want to save this file for history.
- You can get your personal records of distribution in the output file for yourself and for the user on whose behalf you are working. The document description, sender ID, and address are blank. You cannot receive or delete a distribution that is personal when you are working on behalf of another user. The record contains an indicator as to whether the distribution is personal (LINPER and OUTPER).

Character Set and Code Page Considerations

USRID is translated from the value specified in the CMDCHRID parameter to character set 930, code page 500.

The description in the output file is in character set and code page used when the distribution was sent. For the character set and code page, see fields LINCID and LINCPG for incoming distributions and fields OUTCID and OUTCPG for outgoing distributions in the output file.

Receive Distribution (RCVDST) Command

With the Receive Distribution (RCVDST) command, a user can receive incoming distributions, such as documents, notes, and messages.

Files sent by SNDDST TYPE (*FILE) can also be received, but only into an output file. These files are in a distribution output file format, not in the original file format they were in when the SNDDST command was issued to send the data file. Refer to Appendix C, "Office Services Output File Considerations," for more detail.

Incoming mail items can be received to one of the following:

- | | |
|--------------------|--|
| Document | The user specifies the name of the document and folder where the distribution is to be placed (filed) when it is received. This document name must not exist within the specified folder. The document is created as a private document with no authority given to other users. The document is owned by the requester, or by the user who has granted authority to the requester to work on his behalf. |
| Output file | The received distribution can be placed in a database file. See Appendix C, "Office Services Output File Considerations," for more detail. |

When a user sends a distribution with CFMDEL (*YES) and the distribution is received with ACKRCV(*YES), the status is updated on the outgoing distribution of the sender to indicate that the distribution was received.

Security Considerations

You should be aware of the following security considerations:

- You can receive the distributions sent to you.
- You can receive the distributions of another user if you were granted permission to work on behalf of that user (GRTUSRPMN) and the distribution is not personal.
- If attempting to receive the distribution into a folder, the user needs to have *CHANGE authority to the folder.
- If receiving the distribution into an output file, the user needs authority to the output file if it already exists.
- If receiving the distribution into an output file, the user needs authority to the library where the output file is specified.
- If receiving the distribution into an output file that is created, the public authority to the output file is *EXCLUDE.

Some Ways to Use This Command

Following are some ways to use the RCVDST command:

- Use this command to write your own application to receive your distribution.
- To receive distributions from a certain sender, search your QRYDST output file and receive those distributions.

Rules

The following rules apply to this command:

- The distribution identifier is made up of three parts:

The sender's address 8 characters padded with blanks

The sender's user ID 8 characters padded with blanks

The sequence number 4 characters (with leading zeros)

For example, STPAUL MARY 0014 is a distribution ID. STPAUL has two blanks following it, MARY has four blanks following it.

- The distribution extension takes 01 as its default. If you want to receive a distribution where more than one copy was sent to the same recipient, you must specify the value.
- If filing the distribution in the document library, the document you assign cannot exist in the folder.
- A distribution message cannot be filed into the document library.
- When ACKRCV(*YES) is specified, the user who sent the distribution can look at outgoing status to determine whether distributions were delivered.
- KEEP(*NO) is a default. After a distribution is received and optionally acknowledged, the distribution is deleted.
- When KEEP(*YES) is specified, the distribution stays in the log until KEEP(*NO) is specified or the distribution is specifically deleted. The distribution is available for another RCV DST request, or for processing by another document interchange interface.
- If receiving a distribution into a document, the document type is taken from the distribution. If a file was sent that is not in document format, the document type is what was specified when the file was sent. If files are sent that are not in document format, the OfficeVision/400 editor handles these document types as OTHER, and you cannot view or print the documents.

Output File Layout

Refer to Appendix C, "Office Services Output File Considerations," for information on the output file layout for the RCV DST command.

The following record codes are written to the output file, depending on the options taken for the OUTDTATYP parameter:

Figure 4-4 (Page 1 of 2). Record Codes Written to the Output File

| Record Code | Description | OUTDTATYP Value | RCVCRS & RCVCPG Define Data |
|-------------|---------------------------------|--------------------|-----------------------------|
| 010 | Distribution Description record | *DSTINFO *DFT *ALL | Yes |
| 020 | Message Text record | *MSG *DFT *ALL | Yes |
| 105 | Document Description record | *DOCD *DFT *ALL | Yes |
| 110 | Create Date record | *CRTDATE *ALL | No |
| 115 | Expiration Date record | *EXPDATE *ALL | No |
| 120 | Document Date record | *DOCDATE *ALL | No |
| 125 | File Date record | *FILDATE *ALL | No |

Figure 4-4 (Page 2 of 2). Record Codes Written to the Output File

| Record Code | Description | OUTDTATYP Value | RCVCRS & RCVCPG Define Data |
|-------------|--|-----------------|-----------------------------|
| 130 | Date Last Changed record | *CHGDATE *ALL | No |
| 135 | Action Due Date record | *ACTDATE *ALL | No |
| 140 | Completion Date record | *CMPDATE *ALL | Yes |
| 145 | Author records | *AUTHOR *ALL | Yes |
| 150 | Copy List records | *CPYLST *ALL | Yes |
| 155 | Document Class record | *DOCCLS *ALL | Yes |
| 160 | File Cabinet Reference record | *FILCAB *ALL | Yes |
| 165 | Subject records | *SUBJECT *ALL | Yes |
| 170 | Keyword records | *KWD *ALL | Yes |
| 175 | Reference record | *REF *ALL | Yes |
| 180 | Status record | *STATUS *ALL | Yes |
| 185 | Project record | *PROJECT *ALL | Yes |
| 500 | Interchange Document Profile (IDP) records | *IDP *ALL | No |
| 800 | Document Data records | *DOC *DFT *ALL | Yes |

The *OUTDTATYP Value* column tells what values for the OUTDTATYP parameter will create that particular record. The records are created only if that data exists for the record code specified. For example, if Author records are requested and there is no data in the IDP for Author records, then type 145 or Author records in the output file are not created.

The *RCVCRS & RCVCPG Define Data* column tells whether the character set (RCVCRS) and code page (RCVCPG) fields in the record define the data in the data field RCVDTA.

For record type 800, the field RCVDTA contains the data of the distribution. The record code (RCVRC) field defines what kind of data is in the RCVDTA field. The data is stacked in this 500-byte data field. The records will contain 500 bytes, except for the last one, which may be less than 500. For example:

- If a file is received and the record length is 100 bytes, 5 records will fit exactly in this data field.
- If the record length is 80 bytes, 6 full records will use the first 480 bytes and 20 bytes of the next record will make up the remainder of this field. The data in the next RCVDTA record will contain the 60 byte remainder of the data file record, another 5 records, 40 bytes of the following record, and so on until all records have been written.
- If a RFTDCA or FFTDCA document is sent, refer to the *Revisable-Form Text Reference* and *Final-Form Text Reference* manuals for the format of the data streams.

Hints and Techniques

The distribution identifier can be retrieved by using the QRYDST command to an output file. This file can be viewed by using the DSPPFM command. The query output file can be used as input to a program and that program can use the distribution identifier in the records as input to RCV DST commands.

If you want only the distribution description, message text, document description, and document data, use the default OUTDTATYP(*DFT) with the RCV DST command.

Character Set and Code Page Considerations

DSTID and USRID are translated from the value specified in the CMDCHRID parameter to character set 930, code page 500.

In the output file, there are two fields in the record that define what character set and code page the data is in. These fields are RCVCRS and RCVCPG. See the record code table for the record codes where the character set and code page are set for the data.

The IDP record has the character set and code page in the record. See the *Interchange Document Profile Reference* manual for details.

Send Document (SNDDOC) Command

The Send Document (SNDDOC) command displays the Send a Document display from which you send documents to one or more users if OfficeVision/400 is not active.

Security Considerations

You must have *USE authority to the document you are sending.

Some Ways to Use This Command

The SNDDOC command displays the Send a Document display on which the user enters the document to be sent, the folder in which it resides, and the addresses to which the document is to be sent. There are two parameters associated with the SNDDOC command: document name and folder name. Values entered as input parameters for the document and folder names appear on the Send a Document display in their corresponding fields.

Users are able to integrate the distribution of documents with their own user-written applications. That is, SNDDOC can be grouped with other commands and compiled into a CL program.

SNDDOC can be entered on the command entry display, or from a command line in an interactive environment.

SNDDOC is very useful if you only want to send a few documents to some users, either by distribution list, individual addresses, or both.

Rules

The following rules apply to SNDDOC:

- You must be enrolled in OfficeVision/400. Either the security officer or an OfficeVision/400 administrator can enroll users.
- Since SNDDOC presents a display to the user, it cannot be used in a batch environment. It is only available interactively from a command line, or from a CL program.
- Only one document can be sent at a time. However, once the document has been sent, you can enter the information to send another document without exiting from the Send a Document display.
- The Attention key is not active with this command, as it is with the STROFC command. However, your application can do an attention function with the SETATNPGM and TFRGRPJOB commands.
- You cannot use the SNDDOC command if OfficeVision/400 is already active.

Hints and Techniques

Following are some hints and techniques for using SNDDOC:

- Displays and function keys are provided to select folders, documents, distribution lists, and address entries. This is helpful when you want to send a document but you do not know the exact name of the document, folder, distribution lists, or addressees.
- You can change details such as document description, subject, keywords, authors, and so on by using F11.
- A memo slip can be attached to the document by using F9.
- The defaults that confirm delivery, personal, and high priority can be changed by using F13. This is in effect until you exit the send function. You can change the defaults permanently by using F17.

Send Distribution (SNDDST) Command

The Send Distribution (SNDDST) command allows a user to send different types of distributions (a message, document, file, or interchange document profile) to a user, a list of users, or by a distribution list. Some of these distributions can also be sent as a note or as final-form text (FFT).

You can send the following items with the SNDDST command:

Message only When sending a message, the message text is contained in the message parameter (MSG). You can also resend a message by using DSTID, if you specify *DSTIDMSG as a value.

Document with optional message

When sending a document, the document can be sent either by document and folder name, or by DOCID (library-assigned document name). The DOCID can be found by doing a QRYDOCLIB command. The DOCID is also returned in a message when a FILDOC is done. It is also possible to send the document as a note or as final-form text.

Database file with optional message

A database file can be sent also. The file can be a data file with no office considerations (not in office format). In this case, the

data is read and sent within the office network. The data in the database file is put into a distribution.

The file can also be a data file that is formatted to be the same as the output file that the RCV DST or RTV DOC command creates. When the file is in this form, the SNDDST command recognizes that and properly processes the file to format the data and the interchange document profile (IDP). The only reason to use a RCV DST or a RTV DOC command is to change the data and resend it. You could resend a distribution by specifying *DSTID or send a document by specifying *DOC with the TYPE parameter. You can also send the distribution as a note.

You may use the RTV DOC command to copy a document into a database file and send the file using the SNDDST command. It is also possible to write your own application to format a data file in the same manner as does the RTV DOC command and use that file as input to the SNDDST command.

IDP data with optional message

Send the IDP from an output file from the RCV DST or RTV DOC commands. It can also be the IDP data specified by the IDP parameters FILCAB, DOCD, AUTHOR, DOCCLS, KWD, and so on. The FILCAB parameter is required. If no document or data file is sent, the IDP could be describing some other object, such as a printed document. Also, it is possible to write your own application to format a data file for the IDP data and use that file as input to the SNDDST command.

Mail log entries with optional message

The mail-log entry can be distributed. When sending mail log entries, you can send a distribution without receiving it. Specify the distribution identifier, and a copy of the distribution is sent to the user(s) specified.

You can send a message with a document, database file, IDP data, and mail log entry by using the MSG parameter.

Security Considerations

If sending a document, the user must have at least *USE authority to the document, or be working on behalf of another user who has at least *USE authority. Use the GRTUSRPMN command to give authority to work on behalf of another user.

If sending a file or the IDP from a file, the user must have *USE authority to the file.

Some Ways to Use This Command

Following are some ways to use the SNDDST command:

- To delegate means to send the distributions that you receive to someone else to take care of. For the delegate function, you resend the distribution to the user who the distributions are being delegated to. Refer to “Delegating Mail” on page 4-40 for more information on how to do this.
- You can distribute something to specific users who meet certain requirements without having to set up a specific distribution list for this.

Rules

Following are rules that apply to this command:

- When sending mail as a note, you must specify *DOC, *FILE, or *DSTID on the TYPE parameter. The document or data in the file must be in a format that can be changed to final-form text, for example, revisable-form text (RFT).
- When sending mail as final-form text, you must specify *DOC on the TYPE parameter.
- When sending mail as final-form text or as a note, you cannot specify *MSG or *IDP on the TYPE parameter.
- If you want to specify your own document type, use the DOCTYPE parameter and the SYSCOD parameter. These parameters are required if a number 32 768 through 65 535 is specified for the DOCTYPE parameter.
- Up to 300 user IDs or distribution list IDs can be specified for the parameter TOUSRID.
- The requester must be enrolled in the system distribution directory to send distributions.
- If sending a document, the document must exist in the folder specified. If sending a file, the file must exist in the library specified.
- When resending a mail log entry, the distribution ID must be specified. The extension number takes the default value of 01.
- When the sender wants to resend a distribution that was sent with confirm delivery (COD), you must specify 00 as the distribution ID extension.
- For the recipient, the distribution ID is 01 for the first copy of the distribution, and increases by 1 for each duplicate copy thereafter. You should specify the distribution ID extension in a CL program and get the value from the output file.

Hints and Techniques

Following are some hints and techniques for using the SNDDST command:

- Use the WRKDSTQ command to work with the distribution queues after the SNDDST command is done to remote users. This command shows the status of the distribution queues and the position of your distribution.
- When sending a distribution to a remote system, use DSPLINSTS of QSNADS to see if the communication lines are operating.
- If you want to make sure that a distribution is seen by a recipient, use the CFMDEL(*YES) option.
- You can get the distribution identifier from the second level text in the completion message issued after the SNDDST command. If doing this interactively, press the Help key on the message to view the second level text.
- If you are sending a file with the SNDDST command, be aware that the file will be in distribution output file format when it is received, instead of the file format it was in before you sent it. Use SNDNETF or SNDF to send a file (not a distribution file) and receive it in the same file format.
- There are two ways to get a document into an output file:
 - Use RTVDOC.
 - Use SNDDST TYPE(*DOC) and do a RCVDST to an output file.

These output files can be manipulated. To get the data in the file back into a document, use FILDOC or RPLDOC.

Character Set and Code Page Considerations

The following parameters are translated from the value specified in the CMDCHRID parameter to character set 930, code page 500:

- TOUSRID
- USRID
- DSTID

The following parameters are translated from the value specified in the CMDCHRID parameter to character set 697, code page 500:

- SYSCOD
- MSG
- DSTD

The following parameters are not translated and are passed in the character set and code page specified in CMDCHRID:

- DOCD
- AUTHOR
- DOCCLS
- KWD
- SUBJECT
- FILCAB
- CPYLST
- REFERENCE
- STATUS
- PROJECT

Example Applications

This section provides examples of the document distribution services command interfaces. These examples are not meant to be full-function solutions. Instead, they are suggestions for using the commands. Not all programming considerations or techniques are illustrated in these examples. You should review the examples before you begin applications design and coding.

Granting User Permission to All User Profiles in the Directory

Some of the CL program examples in this section require USER authority to the user profile. The example program, GRTALLPMN, shown in Figure 4-5 on page 4-31, allows you to get USER authority automatically.

In the following example, a DSPDIR is done to an output file. The output file is read sequentially. If a user profile name is in the directory entry, the user ID that was input to the program is granted USE authority to the user profile in the directory entry. A message indicates the user profile has been granted permission. A DLTF of the DSPDIR output file is done at the end of the program.

If you run this program several times, performance improves if you do the following:

- Periodically do the DSPDIR to the output file QTEMP/DSPDIR DETAIL(*FULL) outside the program for updates.

- Comment out the DSPDIR and the DLTF commands in the program.

```

PGM          PARM(&USRP)
/* This program grants all users in the system */
/* distribution directory user permission to the */
/* user specified. Be careful in using this program */
/* and give this all permission to those who need it. */
/* */
/* Need to have security officer authority to run */
/* this program. */
/* */
/* The program does a DSPDIR to an outfile. It then */
/* grants user permission to the user specified on */
/* input (&USRP) to each user profile in the */
/* directory. */
/* */
/* For example if you want to grant all users */
/* permission to user profile CHUCK, you would issue */
/* the following command: */
/*   CALL GRTALLPMN PARM(CHUCK) */
/* */
/* To create this program, use the CRTCLPGM command. */
/* This is the source code for the program. */
/* */

DCL          VAR(&USRP) TYPE(*CHAR) LEN(10)
DCL          VAR(&SAVUSRID) TYPE(*CHAR) LEN(8) +
              VALUE(' ') /* Initialize the +
                           saved user ID to blanks */
DCL          VAR(&SAVADDR) TYPE(*CHAR) LEN(8) +
              VALUE(' ') /* Initialize the +
                           saved address to blanks */
DCL          VAR(&KEY) TYPE(*CHAR) LEN(7) /* Message +
                           key value to remove from message queue */

DCLF         FILE(QSYS/QAOSDIRO) RCDfmt(OSDIRE) /* System +
                                                DSPDIR outfile layout */

```

Figure 4-5 (Part 1 of 2). Granting User Permission to All User Profiles in Directory

```

/* Do a DSPDIR to an outfile in library QTEMP          */
DSPDIR      OUTPUT(*OUTFILE) +                          */
            OUTFILE(QTEMP/DSPDIR) DETAIL(*FULL)

/* Now override the system file with the file just    */
/* created by DSPDIR.                                  */
OVRDBF      FILE(QAOSDIRO) TOFILE(QTEMP/DSPDIR)

/* Read the records sequentially from the DSPDIR      */
/* outfile.                                           */
READF:      RCVF      RCD_FMT(OSDIRE)

/* Monitor for EOF message                            */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* If the user profile name is blank, then skip this */
/* entry.                                             */
IF          COND(&WOSDUSR *EQ ' ' ) +
            THEN(GOTO CMDLBL(READF))

/* If this entry is a multiple description entry then */
/* skip it (User ID and Address are the same as the  */
/* previous entry).                                   */
IF          COND(&WOSDDEN *EQ &SAVUSRID +
            *AND &WOSDDGN *EQ &SAVADDR) +
            THEN(GOTO CMDLBL(READF))

/* Set the saved user ID and address to current entry */
CHGVAR      VAR(&SAVUSRID) VALUE(&WOSDDEN)
CHGVAR      VAR(&SAVADDR) VALUE(&WOSDDGN)

/* Grant the user &USRP permission to this users    */
/* profile in the directory.                          */
GRTUSRPMN   TOUSER(&USRP) FORUSER(&WOSDUSR)

/* Monitor for CPF9007 escape message - User         */
/* permission given to N users and not to X users.   */
MONMSG      MSGID(CPF9007) EXEC(GOTO CMDLBL(NEXTR))

/* Send message that says which user granted         */
/* permission to if successful.                       */
SENDPGMMSG  MSG('Granted User Permission to user ' +
            *CAT 'profile ' *CAT &WOSDUSR) +
            MSGTYPE(*INFO)

/* Read another record from the DSPDIR outfile.      */
NEXTR:      GOTO      CMDLBL(READF)

EOF:        /* Delete the temporary DSPDIR outfile.   */
DLTF        FILE(QTEMP/DSPDIR)

ENDPGM

```

Figure 4-5 (Part 2 of 2). Granting User Permission to All User Profiles in Directory

Sending Messages, Notes, and Documents to Office Users

The examples in this section illustrate some uses of sending distribution with commands, and how the commands can fit into your application and computer system.

Example 1

You can send all of the documents from your folder to another user with a CL program. This program can be called by the CALL statement on the CL command entry display, or you can create a command to call the program.

The following OS/400 office commands are used in this example:

DSPFLR The Display Folder command writes all documents in the specified folder to an output file.

SNDDST The Send Distribution command sends a document in the folder to the specified user ID and address. The command also sends a message to the user indicating the document was sent.

The following program, SNDALLDOC, sends the documents:

```
PGM          PARM(&FLR &USRID &ADDR)
/* This program will send all the documents in the */
/* folder specified by &FLR to the user ID specified */
/* by &USRID &ADDR. This user ID and address need */
/* to exist in the system distribution directory and */
/* the folder also needs to exist.                */
/*                                                */
/* Example call of this program from the command */
/* line would be:                                */
/*   CALL SNDMAIN PARM(DOCFLR XYZ DISOSS)        */
/*                                                */
/* where DOCFLR is the name of the folder the */
/* documents to be sent are in and             */
/* XYZ DISOSS is the User ID of the remote user */
/* on the DISOSS system.                        */
/*                                                */
/* To compile this program use the CRTCLPGM command. */
/*                                                */
DCL          VAR(&FLR) TYPE(*CHAR) /* Variable for +
           the folder parameter.                */
DCL          VAR(&USRID) TYPE(*CHAR) /* Variable +
           for the User ID parameter             */
DCL          VAR(&ADDR) TYPE(*CHAR) /* Variable +
           for the Address parameter            */
DCLF         FILE(QSYS/QADSPDOC) RCDFMT(DOCDTL) /* File +
           containing documents to be sent      */
```

Figure 4-6 (Part 1 of 2). Sending Documents

```

/* Monitor for OFC8006 message - In case the folder */
/* does not exist. */
MONMSG      MSGID(OFC8006) EXEC(GOTO CMDLBL(END))

/* Display the folder to an outfile */
DSPFLR      FLR(&FLR) TYPE(*DOC) OUTPUT(*OUTFILE) +
            OUTFILE(QTEMP/OUTFILE) /* List all +
            documents in a folder to an outfile */

/* Override the system database file for the DSPFLR */
/* outfile. Change the file to the outfile from the */
/* DSPFLR CL command. */
OVRDBF      FILE(QADSPDOC) TOFILE(QTEMP/OUTFILE)

/* Now open the DSPFLR outfile */
OPNDBF      FILE(QSYS/QADSPDOC) OPTION(*INP)

/* Read all the records from the DSPFLR outfile */
/* until end of file is reached */
READF:      RCVF      RCDfmt(DOCDTL)

/* Monitor for EOF message */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* Send the document to the user ID */
SNDDST      TYPE(*DOC) TOUSRID((&USRID &ADDR)) +
            DOC(&DOCNAM) FLR(&FLRNAM) +
            DSTD(&DOCNAM)

/* Read another record from the outfile */
GOTO        CMDLBL(READF)

/* Close and delete the outfile that was created */
/* by the DSPFLR command. */
EOF:        CLOF      OPNID(QADSPDOC)
            DLTF      FILE(QTEMP/OUTFILE)

/* Send a message to the user ID */
SNDDST      TYPE(*MSG) TOUSRID((&USRID &ADDR)) +
            DSTD('Documents all sent') +
            MSG('All documents sent from folder' *CAT +
            &FLRNAM)

END:        RETURN    /* Return control to the calling program. */
            ENDPGM

```

Figure 4-6 (Part 2 of 2). Sending Documents

Example 2

If you want to send only documents that meet certain requirements, you can query a folder and send a list of documents to an output file using the QRYDOCLIB command. By using the output file as input to the following program, you can send the documents that match your requirements to the specified user ID and address.

The following program, SNDQRYDOC, sends the documents that meet your requirements:

```

PGM          PARM(&FILENAM &FILELIB &USRID &ADDR)
/* This program will send all the documents in a      */
/* QRYDOCLIB outfile to a user. The outfile is      */
/* specified by the &FILELIB and &FILENAM parameters */
/* and the user ID is specified by the &USRID and    */
/* &ADDR parameters.                                */
/*                                                    */
/* For example, if you have a QRYDOCLIB outfile     */
/* called OUTFILE in library MYLIB and you want to  */
/* send these documents to user CHARLIE at address  */
/* CHICAGO you would issue the following command:   */
/* CALL SNDQRYDOC PARM(MYLIB OUTFILE CHARLIE CHICAGO) */
/*                                                    */
/* To create this program, use the CRTCLPGM command. */
/* This is the source code for the program.         */
/*                                                    */
/* The command SNDQDOC calls this program.          */
/*                                                    */

DCL          VAR(&FILENAM) TYPE(*CHAR) LEN(10) /* +
              Variable for the QRYDOCLIB outfile name */
DCL          VAR(&FILELIB) TYPE(*CHAR) LEN(10) /* +
              Variable for the QRYDOCLIB outfile library*/
DCL          VAR(&USRID) TYPE(*CHAR) LEN(8) /* +
              Variable for the User ID parameter      */
DCL          VAR(&ADDR) TYPE(*CHAR) LEN(8) /* +
              Variable for the Address parameter      */
DCLF         FILE(QSYS/QAOSIQDL) RCDfmt(OSQDL) /* System +
              QRYDOCLIB outfile layout                */

/* Use the outfile from the QRYDOCLIB that was      */
/* input as the file to read the records from.     */
OVRDBF      FILE(QAOSIQDL) TOFILE(&FILELIB/&FILENAM)

READF:      RCVF          RCDfmt(OSQDL) /* Read records from the +
              outfile until the EOF is reached.      */

/* Monitor for EOF message                          */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* Send the document to the user ID                  */
SNDST      TYPE(*DOC) TOUSRID((&USRID &ADDR)) +
              DOC(&QDLDNM) FLR(&QDLFLR) +
              DSTD('Document from DISOSS folder') +
              /* Send each document in folder        */

/* Read another record from the outfile              */
GOTO      CMDLBL(READF)

EOF:       RETURN      /* Return control to the calling program. */
ENDPGM

```

Figure 4-7. Sending Selected Documents

Following is the command source for calling the program, SNDQRYDOC:

```
/*
/* *****
/* COMMAND NAME: SNDQDOC
/*
/* COMMAND TITLE: Send Query Documents
/*
/* DESCRIPTION: This command will send all documents from a
/*                QRYDOCLIB outfile to a user.
/*
/* To create this command, use the CRTCMD command. This is the
/* source file for the command. Call the program SNDQRYDOC
/* from the command (this is specified on the CRTCMD command).
/*
/* *****
SNDQDOC: CMD  PROMPT('Send Query Documents')
           PARM  KWD(FILENAM)  +
                TYPE(*NAME) LEN(10) RSTD(*NO) MIN(1) MAX(1) +
                VARY(*NO) EXPR(*YES) PASSATR(*NO) FULL(*NO) +
                FILE(*IN)  +
                PROMPT('Outfile name from QRYDOCLIB')
           PARM  KWD(FILELIB)  +
                TYPE(*NAME) LEN(10) RSTD(*NO) MIN(1) MAX(1) +
                VARY(*NO) EXPR(*YES) PASSATR(*NO) FULL(*NO) +
                PROMPT('Library outfile is in')
           PARM  KWD(USRID)   +
                TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) MAX(1) +
                EXPR(*YES) FILE(*NO) VARY(*NO) PASSATR(*NO) +
                PROMPT('User ID')
           PARM  KWD(ADDR)   +
                TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) MAX(1) +
                EXPR(*YES) FILE(*NO) VARY(*NO) PASSATR(*NO) +
                PROMPT('User Address')
```

Figure 4-8. Source for Calling SNDQRYDOC

Receiving Messages, Notes, and Documents from a User's Mailbox

This example illustrates how to receive mail with commands and how commands can fit into your application and computer systems.

Run this example program to automatically receive a user's distribution from the mailbox. The program receives messages into a file called MSGFILE in the library of the user ID. Documents of type RFTDCA and FFTDCA are received into the document with the name DOC concatenated with the distribution ID number and placed in the folder of the user ID. Other types of documents are received into a file with the name FILE concatenated with the distribution ID number and placed in the library of the user ID.

This example uses the following OS/400 office commands:

QRYDST The Query Distribution command queries a user's mailbox to get the incoming distribution entries to the output file QTEMP/QRYDSTIN.

RCVDST The Receive Distribution command receives a message into a file, a RFTDCA or FFTDCA type document into a document, and other types of documents into a file.

The following program, RCVMAIL, receives mail:

```
PGM          PARM(&USRID &ADDR)
/* This program will receive all the incoming */
/* distributions sent to the user specified by &USRID */
/* and &ADDR. If the distribution is a message, then */
/* the messages will be put into file MSGFILE of */
/* library &USRID. If the distribution is a document,*/
/* then it will be put into document of DOC + dist id */
/* in folder &USRID. If the distribution is a file */
/* (not RFT or FFT type) then the distribution will */
/* be put in file FILE + dist id in library &USRID. */
/* */
/* The program does a QRYDST of the &USRID and &ADDR */
/* to an outfile. It processes each record in the */
/* QRYDST outfile and does a RCV DST to receive each */
/* distribution. */
/* */
/* For example if you want to receive all of the */
/* distributions sent to user CHARLIE at address */
/* CHICAGO, you would issue the following command: */
/* CALL RCVMAIL PARM(CHARLIE CHICAGO) */
/* */
/* To create this program, use the CRTCLPGM command. */
/* This is the source code for the program. */
/* */
DCL          VAR(&USRID) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADDR) TYPE(*CHAR) LEN(8)
DCL          VAR(&FILENAM) TYPE(*CHAR) LEN(8)
DCL          VAR(&DOCNAME) TYPE(*CHAR) LEN(7)
DCLF        FILE(QSYS/QAOSILIN) RCDFMT(OSLIN) /* System +
          QRYDST outfile layout */
/* Do a QRYDST to an outfile in library QTEMP */
QRYDST      USRID(&USRID &ADDR) OUTFILE(QTEMP/QRYDSTIN)

/* Now override the system file with the file just */
/* created by QRYDST. */
OVRDBF     FILE(QAOSILIN) TOFILE(QTEMP/QRYDSTIN)

/* Read the records sequentially from the QRYDST */
/* outfile. */
READF:     RCVF          RCDFMT(OSLIN)

/* Monitor for EOF message */
MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
```

Figure 4-9 (Part 1 of 2). A Program to Receive Mail

```

/* If the distribution is a message, then receive the */
/* message into a file called MSGFILE in library      */
/* &USRID.                                           */
IF COND(&LINDTP *EQ 0) THEN( +
DO)
    RCV DST DSTID(&LINDID) DSTIDEXN(&LINDEX) +
        USRID(&LINRUI &LINRUA) DOC(*NONE) +
        OUTFILE(&USRID/MSGFILE) +
        OUTMBR(*FIRST *ADD) OUTDTATYP(*ALL)
ENDDO
ELSE +
DO
/* Otherwise if the distribution is a document then */
/* receive it as document 'DOC' concatenated with  */
/* the distribution ID number.                       */
IF COND( (&LINDTP *EQ 2) *OR (&LINDTP *EQ 11) ) +
THEN( DO )
    CHGVAR VAR(&DOCNAME) VALUE( +
        'DOC' *CAT %SST(&LINDID 17 4) )
    RCV DST DSTID(&LINDID) DSTIDEXN(&LINDEX) +
        USRID(&LINRUI &LINRUA) +
        DOC(&DOCNAME) +
        FLR(&USRID)
    MONMSG MSGID(CPF9098) +
        EXEC(GOTO CMDLBL(READF))
ENDDO
ELSE +
DO
/* Otherwise, the distribution is a distribution    */
/* of document type that is not RFT or FFT. Could  */
/* store these in a document if you choose but they */
/* cannot be accessed through OfficeVision/400.    */
/* So they are put into file 'FILE' concatenated   */
/* with the distribution id number.                 */
CHGVAR VAR(&FILENAM) VALUE( +
    'FILE' *CAT %SST(&LINDID 17 4) )
RCV DST DSTID(&LINDID) USRID(&LINRUI &LINRUA) +
    OUTFILE(&USRID/&FILENAM) DSTIDEXN(&LINDEX) +
    OUTMBR(*FIRST *ADD) OUTDTATYP(*ALL)
MONMSG MSGID(CPF9098) +
    EXEC(GOTO CMDLBL(READF))
ENDDO
ENDDO

/* Read the next record in the query file.          */
GOTO      CMDLBL(READF)

EOF:      /* Delete the temporary QRYDST outfile.    */
DLTF      FILE(QTEMP/QRYDSTIN)

ENDPGM

```

Figure 4-9 (Part 2 of 2). A Program to Receive Mail

Delegating Mail

Using the delegate mail function, a user can specify that another user handle his mail. The function starts when the user does the following:

- Runs the Delegate command to specify the user to whom mail is delegated
- Specifies the start and end dates
- Specifies a message to be sent as a reply to the sender

When mail arrives, it is forwarded to the specified user, and a reply is sent to the sender. The delegate function can be ended by running the CNLDELEGAT command. It ends automatically when the end date is reached.

The following OS/400 office commands are used in this application:

QRYDST The Query Distribution command writes a record to a database file for each incoming distribution for a user.

SNDDST The Send Distribution command sends messages to the users. It is also used to send distributions again to the specified user.

RCVDST The Receive Distribution command receives a distribution into a file. The file is distributed to the specified user.

Assumptions

This application includes little error handling capability. It is assumed that in its development environment, the margin for error is small. Therefore, the responsibility for extensive error handling routines lies with the person installing the application.

In addition, no efforts were made to ensure that data security is inherent in the application. The application allows a user to specify any other user as the person delegating, or being delegated to. The responsibility for taking security measures falls to the person installing the application.

The application uses a base file (where the names of users delegating mail are kept), and three other files for each of these users. Following is the record layout of these files, and a brief explanation of their use:

DELEGATE The DELEGATE file contains a record for each of the users using the mail delegate function. Each record contains the following:

- The user ID and address of the person who is away and delegating mail
- The user ID and address of the person handling the mail for the delegating user
- The date when mail starts being forwarded
- The date forwarding stops
- The response sent to senders of mail

The file is keyed on user ID.

Following is the record format for the file, DELEGATE:

```
A** DDS OBJECT:      DELEGATE
A** SOURCE FILE:     QDDSSRC
A** SOURCE LIB:      OFCLIB
A** DESCRIPTION:     LIST OF USERS WHO HAVE DELEGATED THEIR MAIL
A*
A          R DGT
A          USR          8A
A          ADD          8A
A          DGTUSR       8A
A          DGTADD       8A
A          STRDAT       6A
A          STPDAT       6A
A          MSG          256A
A          K USR
```

NEWQRY The NEWQRY file contains the latest list of incoming mail for a user. Each user has one of these files. Because it must be unique to that user, the file name is created by concatenating the user ID with the letters NW.

For example, if the user ID is SMITH the new query file is SMITHNW. The file is the output of a QRYDST command, so the record format name and structure must correspond to that specified by the command (see Appendix C, "Office Services Output File Considerations," for more information).

Following is the record format for the file, NEWQRY:

```
A** DDS OBJECT:      NEWQRY
A** SOURCE FILE:     QDDSSRC
A** SOURCE LIB:      OFCLIB
A** DESCRIPTION:     OUTPUT FILE FOR QRYDST COMMAND IN DELEGATE
A*
A          R OSLIN
A          LINSNO       6S
A          LINDID      20A
A          LINRUI       8A
A          LINRUA       8A
A          LINSUI       8A
A          LINSUA       8A
A          LINSDT       8A
A          LINSTM       6A
A          LINPER       1A
A          LINPTY       1A
A          LINCOD       1A
A          LINDTP       5S
A          LINSYC      13A
A          LINCID       3S
A          LINCPCG      3S
A          LINDSD      44A
A          LINDEX       2S
A          LINMLS       1A
A          LINCNT       8A
A          LINCTM       6A
A          K LINDID
```

OLDQRY The OLDQRY file contains the last list of the incoming mail. Before the new listing is created and stored in NEWQRY, the contents of NEWQRY are copied to OLDQRY. OLDQRY contains the previous, and the latest, list. Each user must have his own OLDQRY file. Therefore, the name of the file is created by concatenating the user ID with the letters OL. For example, SMITH would have an old query file named SMITHOL.

Following is the record format for the file, OLDQRY:

```

A** DDS OBJECT:      OLDQRY
A** SOURCE FILE:    QDDSSRC
A** SOURCE LIB:     OFCLIB
A** DESCRIPTION:    FILE FOR OLD QRYDST LISTING
A*
A      R  OQRY
A      LINSNO      6S
A      LINDID     20A
A      LINRUI      8A
A      LINRUA      8A
A      LINSUI      8A
A      LINSUA      8A
A      LINSDT      8A
A      LINSTM      6A
A      LINPER      1A
A      LINPTY      1A
A      LINCOD      1A
A      LINDTP      5S
A      LINSYC     13A
A      LINCID      3S
A      LINCPG      3S
A      LINDSD     44A
A      LINDEX      2S
A      LINMLS      1A
A      LINCDD      8A
A      LINCTM      6A
A      K  LINDID

```

ADDMAIL The records in file ADDMAIL are created when the NEWQRY and OLDQRY files are compared. Any new records in NEWQRY that are not in OLDQRY are assumed to be new mail items, and are placed in ADDMAIL.

Following is the record format for the file, ADDMAIL:

```
A** DDS OBJECT:      ADDMAIL
A** SOURCE FILE:     QDSSRC
A** SOURCE LIB:      OFCLIB
A** DESCRIPTION:     ADDITIONAL MAIL FILE AFTER COMPARE IN DELEGATE
A*
A          R ADDM
A          LINSNO      6S
A          LINDID     20A
A          LINRUI      8A
A          LINRUA      8A
A          LINSUI      8A
A          LINSUA      8A
A          LINSDT      8A
A          LINSTM      6A
A          LINPER      1A
A          LINPTY      1A
A          LINCOD      1A
A          LINDTP      5S
A          LINSYC     13A
A          LINCID      3S
A          LINCPCG     3S
A          LINDSD     44A
A          LINDEX      2S
A          LINMLS      1A
A          LINCNT      8A
A          LINCTM      6A
A          K LINDID
```

Code

The application is divided into the following sections:

- The call of the mail delegate function
- Periodic checking of mail for users
- Ending the function

The library name OFCLIB is used to keep the programs and files.

Mail Delegate Code: The mail delegate function is called by the DELEGATE command. This command calls a CL program to start processes like adding the user to the DELEGATE file, and creating the new query, old query, and additional mail files for the user.

Following is the DELEGATE command source:

```

/*****
/*
/* COMMAND NAME: DELEGATE
/*
/* COMMAND TITLE: DELEGATE FUNCTION
/*
/* COMPONENT:
/*
/* DESCRIPTION: This command will allow a user to delegate
/*                responsibility to OTHER users
/*
/*
/*
/*****
DELEGATE:  CMD          PROMPT('Delegate Mail Command')
          PARM          KWD(TOUSRID) TYPE(*CHAR) LEN(8) MIN(1) MAX(1) +
                      RSTD(*NO) EXPR(*YES) VARY(*NO) PASSATR(*NO) +
                      FILE(*NO) PROMPT('To user's user id')
          PARM          KWD(TOADDR) TYPE(*CHAR) LEN(8) MIN(1) MAX(1) +
                      RSTD(*NO) EXPR(*YES) VARY(*NO) PASSATR(*NO) +
                      FILE(*NO) PROMPT('To user's address')
          PARM          KWD(FORUSRID) TYPE(*CHAR) LEN(8) MIN(1) +
                      RSTD(*NO) EXPR(*YES) VARY(*NO) PASSATR(*NO) +
                      MAX(1) FILE(*NO) PROMPT('For user's user +
                      id')
          PARM          KWD(FORADDR) TYPE(*CHAR) LEN(8) MIN(1) MAX(1) +
                      RSTD(*NO) EXPR(*YES) VARY(*NO) PASSATR(*NO) +
                      FILE(*NO) PROMPT('For user's address')
          PARM KWD(STRDATE) +
                      TYPE(*DATE) RSTD(*NO) +
                      MIN(1) MAX(1) +
                      PASSATR(*NO) PROMPT('Start Date')
          PARM KWD(ENDDATE) +
                      TYPE(*DATE) RSTD(*NO) +
                      MIN(1) MAX(1) +
                      PASSATR(*NO) PROMPT('End Date')
          PARM KWD(MSG) +
                      TYPE(*CHAR) LEN(256) RSTD(*NO) +
                      MIN(0) MAX(1) DFT(*NONE) SPCVAL(*NONE) +
                      FILE(*NO) FULL(*NO) EXPR(*YES) VARY(*NO) +
                      PASSATR(*NO) PROMPT('Message')

```

Figure 4-10. DELEGATE Command Source

The command consists of the command definition text and the CL program it calls. The command calls the CL program, QDELEGATE, to start routines for the user. Following is the code for QDELEGATE:

```

/*****
/* THIS PROGRAM WILL ENROLL THE REQUESTING USER IN THE MAIL */
/* DELEGATE FUNCTION, SEND MESSAGE TO THE USER THAT WILL HANDLE */
/* MAIL AND CREATE 3 FILES FOR THE REQUESTING USER */
/*****
PGM PARM(&TOUSRID &TOADDR &FORUSRID &FORADDR &STRDATE &ENDDATE &MSG)
/*****
/* VARIABLE DECLARES */
/*****
DCL &TOUSRID TYPE(*CHAR) LEN(8)
DCL &TOADDR TYPE(*CHAR) LEN(8)
DCL &FORUSRID TYPE(*CHAR) LEN(8)
DCL &FORADDR TYPE(*CHAR) LEN(8)
DCL &STRDATE TYPE(*CHAR) LEN(7)
DCL &ENDDATE TYPE(*CHAR) LEN(7)
DCL &STARTD TYPE(*CHAR) LEN(6)
DCL &STOPD TYPE(*CHAR) LEN(6)
DCL &MSG TYPE(*CHAR) LEN(256)
DCL &NEWQRY TYPE(*CHAR) LEN(10)
DCL &OLDQRY TYPE(*CHAR) LEN(10)
DCL &ADDQRY TYPE(*CHAR) LEN(10)
DCL &DSTD TYPE(*CHAR) LEN(44)
DCL &NONE TYPE(*CHAR) LEN(5) VALUE('*NONE')
DCL &USRFLG TYPE(*CHAR) LEN(1)
/*****
/* REMOVE CENTURY BYTE FROM START AND END DATE THAT WAS PASSED IN */
/* FROM THE COMMAND AND STORE INTO ANOTHER VARIABLE */
/*****
CHGVAR VAR(&STARTD) VALUE(%SST(&STRDATE 2 6))
CHGVAR VAR(&STOPD) VALUE(%SST(&ENDDATE 2 6))
/*****
/* CHECK TO SEE IF THE REQUESTING USER DOES NOT WISH TO SEND A */
/* MESSAGE BACK TO THE PERSON WHO WILL BE HANDLING HIS/HER MAIL */
/*****
IF (&MSG *NE &NONE) THEN+
(DO)
CHGVAR VAR(&DSTD) VALUE(%SST(&MSG 1 44))
SNDDST TYPE(*MSG) TOUSRID((&TOUSRID &TOADDR)) DSTD(&DSTD) MSG(&MSG)
ENDDO
/*****
/* CALL RPG PROGRAM TO ADD DELEGATE USER RECORD TO THE DELEGATE */
/* MAIL FILE AND RETURN A FLAG INDICATING IF THE USER IS ALREADY */
/* ENROLLED OR NOT */
/*****
CALL PGM(OFCLIB/ADDLIST) PARM(&FORUSRID &FORADDR +
&TOUSRID &TOADDR &STARTD &STOPD &MSG +
&USRFLG)

```

Figure 4-11 (Part 1 of 2). QDELEGATE Program

```

/*****
/* CHECK USRFLAG TO SEE IF USER IS ALREADY ENROLLED OR NOT      */
/*****
IF (&USRFLG *EQ 'Y') THEN( +
DO)
    SNDPGMMSG MSG('User already in delegate function, +
                cannot be enrolled more than once')
    GOTO CMDLBL(EXIT)
ENDDO
/*****
/* CREATE 3 FILES TO BE USED LATER TO COMPARE IF NEW MAIL HAS   */
/* ARRIVED YET                                                    */
/*****
CHGVAR VAR(&NEWQRY) VALUE(&FORUSRID *TCAT 'NW')
CHGVAR VAR(&OLDQRY) VALUE(&FORUSRID *TCAT 'OL')
CHGVAR VAR(&ADDQRY) VALUE(&FORUSRID *TCAT 'AD')
CRTPF      FILE(OFCLIB/&NEWQRY) SRCFILE(OFCLIB/QDDSSRC) +
          SRCMBR(NEWQRY)
CRTPF      FILE(OFCLIB/&OLDQRY) SRCFILE(OFCLIB/QDDSSRC) +
          SRCMBR(OLDQRY)
CRTPF      FILE(OFCLIB/&ADDQRY) SRCFILE(OFCLIB/QDDSSRC) +
          SRCMBR(ADMAIL)

EXIT:
ENDPGM

```

Figure 4-11 (Part 2 of 2). QDELEGATE Program

QDELEGATE receives the parameters passed by the command and declares them for local processing. The start and end dates passed by the command are changed to 6-byte character strings. Then QDELEGATE calls the RPG program, ADDLIST, to add the user to the DELEGATE file.

ADDLIST sets a flag if the user is already delegating mail. QDELEGATE checks the flag. If the flag is on, QDELEGATE informs the user that they cannot use the delegate function more than once per time. However, if the flag is not on, QDELEGATE creates the file names for the user and creates the files, which ends the mail delegate start process. The user's mail is forwarded from the time the start date arrives until the end date is reached, or until the user issues the CNLDELEGAT command.

Following is the code for the ADDLIST program:

```

H** RPG OBJECT:          ADDLIST
H** SOURCE FILE:        QRPGRSRC
H** SOURCE LIB:         OFCLIB
H**
H** DESCRIPTION: ADD USERS WHO HAVE DELEGATED THEIR MAIL TO THE
H**                  FILE "DELEGATE"
H**
H** ABNORMAL
H** TERMINATION: RE-RUN THE CL PROGRAM DRIVING THIS PROGRAM
H**              TO CLEAR THE OUTPUT FILES AND TRY AGAIN
H**
H*****
H**                  SUBROUTINE SUMMARY
H*****
H** *PSSR - STANDARD ERROR HANDLING ROUTINE
H*****
H**                  INDICATOR USAGE
H*****
H**    11 - NOT FOUND DELEGATE
H**    97 - NO HIT ON COMPANY HEADER FILE
H*****
H** UPDATE FORMAT FOR THIS PGM YYMMDD - DEBUG=1 FOR DUMP CAPAB
H*****
H      1  Y
F*****
F* DEFINE THE "DELEGATE" FILE
F*****
FDELEGATEIF E          K          DISK          A
F                      KINFSR *PSSR
I*****
IERRDS      SDS
I
I                      *STATUS ESTS1
I                      *ROUTINE ROUT1
I                      *PARMS  PARM1
I                      *PROGRAM PGMNAM
I                      81  90 PGMLIB
I                      244 253 JOBNAM
I                      254 263 USRNAM
I                      264 2690JOBNUM
I                      282 2870JOBTIM
I*
I*

```

Figure 4-12 (Part 1 of 2). ADDLIST Program

```

C*
C*
C*
C*****
C* ADD USER TO DELEGATE FILE
C*****
C*
C      *ENTRY   PLIST
C              PARM      USR      8
C              PARM      ADD      8
C              PARM      DGTUSR   8
C              PARM      DGTADD   8
C              PARM      STRDAT   6
C              PARM      STPDAT   6
C              PARM      MSG      256
C              PARM      USRFLG   1
C*
C      USR      CHAINDGT          11   NOT FOUND
C*
C      *IN11   IFEQ '0'
C              MOVE 'Y'          USRFLG
C*
C              ELSE
C*
C              WRITEDGT
C              MOVE 'N'          USRFLG
C              END
C*
C              MOVE '1'          *INLR
C              RETRN
C*
C*****
C* PSSR          * - PROGRAM ERROR SUBROUTINE
C*****
C* TO DUMP DEBUG MUST BE EQUAL TO 1 (SEE H SPEC)
C*****
C      *PSSR    BEGSR
C      PGMNAM   DUMP
C              MOVE '1'          *INLR
C              RETRN
C              ENDSR

```

Figure 4-12 (Part 2 of 2). ADDLIST Program

Periodic Mail Check Process: Periodic mail checking is the application example. Once a user has enrolled for mail delegate, incoming mail must be checked periodically for new mail, and the new mail must be forwarded to the person handling the mail. The process is accomplished using a main CL program, CHKDGT, and three called programs:

COMPARE An RPG program
SNDMSG A CL program
FORWARD A CL program

Following is the code for the main program, CHKDGT:

```

PGM
DCLF      FILE(OFCLIB/DELEGATE) /* Declare file DELEGATE +
          which contains a list of all users who have +
          delegated their mail. */
DCL      VAR(&NEWQRY) TYPE(*CHAR) LEN(10)
DCL      VAR(&OLDQRY) TYPE(*CHAR) LEN(10)
DCL      VAR(&ADDMAIL) TYPE(*CHAR) LEN(10)
DCL      VAR(&QDATE) TYPE(*CHAR) LEN(6)
DCL      VAR(&QFMT)  TYPE(*CHAR) LEN(3)
DCL      VAR(&FMT)   TYPE(*CHAR) LEN(4)
DCL      VAR(&SYSDAT) TYPE(*CHAR) LEN(6)
RTVSYSVAL SYSVAL(QDATE) RTNVAR(&QDATE) /* Retrieve the +
          system value which is in MMDDYY format. */
RTVSYSVAL SYSVAL(QDATFMT) RTNVAR(&QFMT) /* Retrieve the +
          date format. */
CHGVAR    VAR(&FMT) VALUE('*' *CAT &QFMT)
CVTDAT    DATE(&QDATE) TOVAR(&SYSDAT) FROMFMT(&FMT) +
          TOFMT(*YMD) TOSEP(*NONE) /* Convert date to +
          YYMMDD format */
NEXT:     RCVF      RCDFMT(DGT) /* Read one record at a time from +
          file DELEGATE. */
MONMSG    MSGID(CPF0864) EXEC(GOTO CMDLBL(END))
IF        COND(&STRDAT *GT &SYSDAT) THEN(GOTO +
          CMDLBL(NEXT)) /* If the start date has not +
          yet been reached for this user, don't do +
          anything about his mail and go to the next +
          user. */
CHGVAR    VAR(&NEWQRY) VALUE(&USR *TCAT 'NW')
CHGVAR    VAR(&OLDQRY) VALUE(&USR *TCAT 'OL')
CHGVAR    VAR(&ADDMAIL) VALUE(&USR *TCAT 'AD')
CPYF      FROMFILE(OFCLIB/&NEWQRY) TOFILE(OFCLIB/&OLDQRY) +
          MBROPT(*REPLACE) FMTOPT(*NOCHK)
MONMSG    MSGID(CFP2802) EXEC(SNDPGMMMSG MSG('From file +
          in CPYF command not found') +
          TOMSGQ(Message Queue) MSGTYPE(*INFO))

```

Figure 4-13 (Part 1 of 2). CHKDGT Program

```

MONMSG      MSGID(CPF2861) EXEC(SNDPGMMSG MSG('TO-file in +
              CPYF command not found') +
              TOMSGQ(Message Queue) MSGTYPE(*INFO))
MONMSG      MSGID(CPF2817) EXEC(SNDPGMMSG MSG('FROM-FILE +
              IN CPYF command IS EMPTY') +
              TOMSGQ(Message Queue) MSGTYPE(*INFO))
CLRPFM      FILE(OFCLIB/&ADDMAIL) /* Clear the new mail +
              file for next compare. */
MONMSG      MSGID(CPF3142) EXEC(SNDPGMMSG MSG('File to be +
              cleared not found') TOMSGQ(Message Queue) +
              MSGTYPE(*INFO))
QRYDST      USRID(&USR &ADD) OUTFILE(OFCLIB/&NEWQRY) /* +
              Query the incoming mail for user &USR at +
              &ADD and place the output in file &NEWQRY */
OVRDBF      FILE(NEWQRY) TOFILE(OFCLIB/&NEWQRY) MBR(*FIRST)
OVRDBF      FILE(OLDQRY) TOFILE(OFCLIB/&OLDQRY) MBR(*FIRST)
OVRDBF      FILE(ADDMAIL) TOFILE(OFCLIB/&ADDMAIL) +
              MBR(*FIRST)
CALL         PGM(OFCLIB/COMPARE) /* Call the RPG program to +
              compare the new list to the old list +
              and come up with the new mail items since +
              last check. */
CALL         PGM(OFCLIB/SNDMSG) PARM(&USR &ADD &MSG) /* +
              Call the CL program which will send a +
              message to the sender. */
CALL         PGM(OFCLIB/FORWARD) PARM(&USR &ADD &DGTUSR +
              &DGTADD) /* Call the CL program which will +
              forward the mail to the delegated user. */
IF           COND(&STPDAT *LT &SYSDAT) THEN(CALL +
              PGM(QCNLDLGT) PARM(&USR &ADD)) /* If the +
              stop date has been reached call the program +
              to delete the user from the delegate file +
              and delete the files for that user. */
GOTO        CMDLBL(NEXT) /* Go back and read next record. +
              */
END:        DLTOVR FILE(*ALL)
           ENDPGM

```

Figure 4-13 (Part 2 of 2). CHKDGT Program

CHKDGT starts processing by declaring the file DELEGATE for local use. This file contains one record for each user delegating mail. Since the start date and end date must be compared with the current date, the system current date must be retrieved and converted to the same format as the start and end dates.

Then the first record in DELEGATE is read. The program then knows the following:

- The first user ID and address
- The delegated user ID and address
- The start and end dates

The start date is compared with the current date. If the start date has not been reached, nothing needs to be done for this user. The program branches to get the next record and the next user. If the start date is here, the file names of the three file for this user are created following the naming scheme used when the files were created. See "Assumptions" on page 4-40 for a description of the files and their uses.

The contents of useridNW are copied into useridOL. UseridNW still contains the results of the last QRYDST for this user, but the results are now old so they go to useridOL. The QRYDST command runs, and the output is stored in useridNW. Now there is a new list of incoming mail, and a previous list. If compared, items in the new list that are not in the old list are assumed to have arrived since the last check. To perform this comparison, CHKDGT calls the RPG program, COMPARE.

When COMPARE returns control to CHGDGT, new mail items for this user have been placed in the file USEIDAD. A reply to these items must be sent to the sender of the mail, so CHKDGT calls the CL program, SNDMSG. The new mail must be forwarded to the person handling the mail for the user (a copy of the mail is forwarded and the original is kept in the incoming queue). CHKDGT calls the CL program, FORWARD.

Then CHKDGT compares the end date to the current date to determine whether the end date has arrived. If the end date has been reached, the CL program QCNDLGT is called to remove the user from the delegate function. If the end date has not been reached, the program branches to get the next record and repeats the process for the next user.

Following is the code for the RPG program, COMPARE:

```

H** RPG OBJECT:          COMPARE
H** SOURCE FILE:        QRPGSRC
H** SOURCE LIB:         OFCLIB
H**
H** DESCRIPTION: COMPARE THE NEW OUTFILE OF "QRYDST" AGAINST THE
H**                OLD FILE TO GET NEW MAIL ITEMS SINCE LAST RUN
H**
H** ABNORMAL
H** TERMINATION: RE-RUN THE CL PROGRAM DRIVING THIS PROGRAM
H**                TO CLEAR THE OUTPUT FILES AND TRY AGAIN
H**
H*****
H**                SUBROUTINE SUMMARY
H*****
H** INIT    - PROGRAM INITIALIZATION ROUTINE
H** *PSSR   - STANDARD ERROR HANDLING ROUTINE
H*****
H**                INDICATOR USE
H*****
H**    10 - EOF NEWQRY
H**    11 - NOT FOUND OLDQRY
H**    97 - NO HIT ON COMPANY HEADER FILE
H*****
H** UPDATE FORMAT FOR THIS PGM YYMMDD - DEBUG=1 FOR DUMP CAPAB
H*****
H      1  Y
F*****
F* NEW QUERY OUTPUT FILE
F*****
FNEWQRY  IF  E          K          DISK
F
KINFSR *PSSR

```

Figure 4-14 (Part 1 of 2). COMPARE Program

```

F*****
F* OLD QUERY OUTPUT FILE
F*****
FOLDQRY IF E K DISK
F KINFSR *PSSR
F*****
F* FILE WITH NEW MAIL ITEMS
F*****
FADDMAIL O E DISK
F KINFSR *PSSR
I*****
IERRDS SDS
I *STATUS ESTS1
I *ROUTINE ROUT1
I *PARMS PARM1
I *PROGRAM PGMNAM
I 81 90 PGMLIB
I 244 253 JOBNAM
I 254 263 USRNAM
I 264 2690JOBNUM
I 282 2870JOBTIM
I*
I*
C*
C*
C*
C*****
C* CHECK FOR NEW OR CHANGED RECORDS
C*****
C READ OSLIN 10EOF
C *IN10 DOWEQ'0' EXIT AT EOF
C*
C LINDID CHAINQRY 11 NOT FOUND
C*
C *IN11 IFEQ '1'
C WRITEADDM
C*
C ELSE
C END
C READ OSLIN 10EOF
C END
C MOVE '1' *INLR
C RETRN
C*
C*****
C* PSSR * - PROGRAM ERROR SUBROUTINE
C*****
C* TO DUMP DEBUG MUST BE EQUAL TO 1 (SEE H SPEC)
C*****
C *PSSR BEGSR
C PGMNAM DUMP
C MOVE '1' *INLR
C RETRN
C ENDSR

```

Figure 4-14 (Part 2 of 2). COMPARE Program

COMPARE is called by the main program, CHKDGT, to compare the latest list of incoming mail to the previous list and identify the new mail items.

In the file description specifications, the files named NEWQRY and OLDQRY are identified as externally described input files, and the file named ADDMAIL is defined as an externally described output file. Before CHKDGT calls COMPARE, it overrides the file names with the appropriate user file names so COMPARE processes the appropriate files.

In the calculations specifications, the record format for NEWQRY is read in (READ OSLIN) the first record. Indicator *IN10 is set to on when the end of the file is reached for NEWQRY. Meanwhile, a search is done in OLDQRY for the corresponding record using the distribution ID as the argument (LINDID CHAIN OQRY). If the record is not found, indicator *IN11 is set to on. If the record is not found, the mail item must have arrived since the last check. Therefore, it is written to file ADDMAIL (WRITE ADDM). If the record is found, the next record is read. This process continues until the end of file is reached, and all new mail items have been written to ADDMAIL. Control is passed back to the calling program CHKDGT.

Following is the code for the CL program, SNDMSG:

```

PGM          PARM(&USR &ADD &MSG)
DCLF         FILE(OFCLIB/ADDMAIL) /* Declare file with new +
           mail items. */
DCL          VAR(&USR) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADD) TYPE(*CHAR) LEN(8)
DCL          VAR(&MSG) TYPE(*CHAR) LEN(256)
NEXT:        RCVF         RCDFMT(ADDM) /* Read records from ADDMAIL +
           file one at a time until the end of file is +
           reached. */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(END))
SNDST       TYPE(*MSG) TOUSRID((&LINSUI &LINSUA)) DSTD('I +
           am gone reply') MSG(&MSG) USRID(&USR &ADD) +
           /* Send "I am gone" reply to sender. */
GOTO        CMDLBL(NEXT)
END:         RETURN
            ENDPGM

```

Figure 4-15. SNDMSG Program

SNDMSG is called by the main program, CHKDGT, to reply to the sender of each item of mail in ADDMAIL. The file ADDMAIL is declared for local processing, and the first record is read. The record contains the user ID and address of the sender. Then a message is sent to that user ID and address. The message sent was specified by the user when he issued the Delegate command. The message, user ID, and address are passed as parameters by the calling program. After a message has been sent for all records in ADDMAIL, control is returned to the calling program, CHKDGT.

Following is the code for the CL program, FORWARD:

```
PGM          PARM(&USR &ADD &DGTUSR &DGTADD)
DCLF         FILE(OFCLIB/ADDMAIL) /* Declare file with new +
           mail items. */
DCL          VAR(&USR) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADD) TYPE(*CHAR) LEN(8)
DCL          VAR(&DGTUSR) TYPE(*CHAR) LEN(8)
DCL          VAR(&DGTADD) TYPE(*CHAR) LEN(8)
DCL          VAR(&MSG) TYPE(*CHAR) LEN(44)
CHGVAR       VAR(&MSG) VALUE('Delegated mail from ' *CAT +
           &USR *CAT ' at ' *CAT &ADD)

NEXT:        RCVF          RCD_FMT(ADDM) /* Read records from the file +
           ADDMAIL until the end of file is reached. */
MONMSG       MSGID(CPF0864) EXEC(GOTO CMDLBL(END))
/* If the mail is not personal, then resend it to the */
/* delegated user. */
IF (&LINPER *EQ '0') THEN(DO)
SNDDST       TYPE(*DSTID) TOUSRID((&DGTUSR &DGTADD)) +
           DSTD(&MSG) MSG(*DSTIDMSG) USRID(&USR &ADD) +
           DSTID(&LINDID) DSTIDEXN(&LINDEX)
MONMSG       MSGID(CPF901A)
ENDDO
GOTO CMDLBL(NEXT)
END:         RETURN
ENDPGM
```

Figure 4-16. FORWARD Program

FORWARD is called by the main program, CHKDGT, to route the new, incoming mail items in ADDMAIL to the person handling the mail for the user. ADDMAIL is declared and the first record is read. The mail item is forwarded using the SNDDST command. All non-personal mail records in ADDMAIL are forwarded, and when the end of the file is reached, control is returned to CHKDGT.

Ending Mail Delegate

The mail delegate function can be ended in two ways:

- The user runs the CNLDELEGAT command
- The end date specified in the DELEGATE command is reached

The CNLDELEGATE command consists of the following programs:

- The command description program, CNLDDLGT
- The called program, QCNLDDLGT

Following is the code for the command description program, CNLDDLGT:

```

/*****
/*
/*  COMMAND NAME: CNLDELEGAT
/*
/*  COMMAND TITLE: CANCEL DELEGATE FUNCTION
/*
/*  COMPONENT:
/*
/*  DESCRIPTION: This command will allow a user to cancel
/*                their delegation to another user
/*
/*
*****/
CNLDELEGAT: CMD  PROMPT('Cancel Delegate Command')
               PARM  KWD(FORUSRID) +
                  TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) MAX(1) +
                  EXPR(*YES) FILE(*NO) VARY(*NO) PASSATR(*NO) +
                  PROMPT('For user''s user id')
               PARM  KWD(FORADDR) +
                  TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) MAX(1) +
                  EXPR(*YES) FILE(*NO) VARY(*NO) PASSATR(*NO) +
                  PROMPT('For user''s address')

```

Figure 4-17. CNLDDLGT Program

Following is the code for the CL program, QCNLDLGT:

```

PGM PARM(&RMVUSRID &RMVADDR)
/*****
/* THIS PROGRAM WILL REMOVE A USER FROM THE DELEGATE FUNCTION */
/*****
DCL &RMVUSRID TYPE(*CHAR) LEN(8)
DCL &RMVADDR TYPE(*CHAR) LEN(8)
DCL &TOUSRID TYPE(*CHAR) LEN(8)
DCL &TOADDR TYPE(*CHAR) LEN(8)
DCL &DSTDESC TYPE(*CHAR) LEN(44)
DCL &RMVMSG TYPE(*CHAR) LEN(256)
DCL &FLAG TYPE(*CHAR) LEN(1)
DCL &NEWQRY TYPE(*CHAR) LEN(10)
DCL &OLDQRY TYPE(*CHAR) LEN(10)
DCL &ADDMAIL TYPE(*CHAR) LEN(10)
/*****
/* CALL RPG PROGRAM TO REMOVE USER'S USERID AND ADDRESS */
/*****
CALL PGM(OFLIB/RMVNAME) PARM(&RMVUSRID &RMVADDR +
&TOUSRID &TOADDR &FLAG)
/*****
/* CHECK FLAG TO SEE IF USER WAS IN THE DELEGATE FILE AND WAS */
/* REMOVED FROM THAT FILE */
/*****
IF (&FLAG *EQ 'Y') THEN+
(DO)
/*****
/* CREATE MESSAGE TO SEND TO THE USER (DELEGATEE) WHO WAS */
/* HANDLING MAIL FOR THE USER WHO WAS AWAY(DELEGATOR) */
/*****
CHGVAR VAR(&RMVMSG) VALUE('USER ' *CAT &RMVUSRID +
*CAT ' ' *CAT &RMVADDR *CAT ' IS NO +
LONGER DELEGATING HIS/HER MAIL TO YOU.')
```

Figure 4-18 (Part 1 of 2). QCNLDLGT Program

```

/*****/
/* CHANGE VARIABLE TO GET CORRECT FILE NAMES FOR DELEGATE */
/* USER */
/*****/
      CHGVAR VAR(&NEWQRY) VALUE(&RMVUSRID *TCAT 'NW')
      CHGVAR VAR(&OLDQRY) VALUE(&RMVUSRID *TCAT 'OL')
      CHGVAR VAR(&ADDMAIL) VALUE(&RMVUSRID *TCAT 'AD')
/*****/
/* DELETE ALL THE FILES THAT WERE CREATED */
/*****/
      DLTf FILE(OFCLIB/&NEWQRY)
      DLTf FILE(OFCLIB/&OLDQRY)
      DLTf FILE(OFCLIB/&ADDMAIL)

ENDDO
ENDPGM

```

Figure 4-18 (Part 2 of 2). QCNLDLGT Program

This program ends the delegate function for a user by doing the following:

- Removing his name from the DELEGATE file
- Deleting the three files previously created for the user

When called, the user ID and address of the person to be removed are passed to this program as parameters. Then QCNLDLGT calls the RPG program RMVNAME to remove the user from the DELEGATE file. RMVNAME returns a flag if the user is found. QCNLDLGT checks the flag. If the flag is on (meaning the user was found and removed), the files are removed. If the flag indicates the user was not found, control is returned to the calling program.

Following is the code for the RPG program, RMVNAME:

```
H** RPG OBJECT:          RMVNAME
H** SOURCE FILE:        QRPGRSRC
H** SOURCE LIB:         OFCLIB
H**
H** DESCRIPTION: REMOVE USERS FROM THE DELEGATE FILE TO END THE
H**                  DELEGATE FUNCTION
H**
H** ABNORMAL
H** TERMINATION: RE-RUN THE CL PROGRAM DRIVING THIS PROGRAM
H**                  TO CLEAR THE OUTPUT FILES AND TRY AGAIN
H**
H*****
H**                  SUBROUTINE SUMMARY
H*****
H** *PSSR - STANDARD ERROR HANDLING ROUTINE
H*****
H**                  INDICATOR USE
H*****
H** 11 - NOT FOUND DELEGATE
H** 97 - NO HIT ON COMPANY HEADER FILE
H*****
H** UPDATE FORMAT FOR THIS PGM YYMMDD - DEBUG=1 FOR DUMP CAPAB
H*****
H      1  Y
F*****
F* DEFINE THE "DELEGATE" FILE
F*****
FDELEGATEUF E          K          DISK
F
F                                KINFSR *PSSR
I*****
IERRDS      SDS
I
I                                *STATUS ESTS1
I                                *ROUTINE ROUT1
I                                *PARMS  PARM1
I                                *PROGRAM PGMNAM
I                                81  90 PGMLIB
I                                244 253 JOBNAM
I                                254 263 USRNAM
I                                264 2690JOBNUM
I                                282 2870JOBTIM
I*
I*
```

Figure 4-19 (Part 1 of 2). RMVNAME Program

```

C*
C*
C*
C*****
C* REMOVE USER FROM DELEGATE FILE
C*****
C*
C      *ENTRY  PLIST
C          PARM      USR      8
C          PARM      ADD      8
C          PARM      USR2     8
C          PARM      ADD2     8
C          PARM      FLAG     1
C*
C      USR      CHAINDGT              11  NOT FOUND
C*
C      *IN11   IFEQ '0'
C          MOVE DGTUSR  USR2
C          MOVE DGTADD  ADD2
C          MOVE 'Y'     FLAG
C          DELETEDGT
C*
C          ELSE
C          MOVE 'N'     FLAG
C*
C          END
C*
C          MOVE '1'     *INLR
C          RETRN
C*
C*****
C* PSSR      * - PROGRAM ERROR SUBROUTINE
C*****
C* TO DUMP DEBUG MUST BE EQUAL TO 1 (SEE H SPEC)
C*****
C      *PSSR  BEGSR
C      PGMNAM DUMP
C          MOVE '1'     *INLR
C          RETRN
C          ENDSR

```

Figure 4-19 (Part 2 of 2). RMVNAME Program

QCNDLGT calls RMVNAME to remove the name of a user from the DELEGATE file. In the file description specifications, the DELEGATE file is defined as an externally described file for input and update. In the calculations specifications, the parameters passed by the calling program are declared for local processing (the *ENTRY PLIST statement). Then the record for the user is searched using user ID as the argument (USR CHAIN DGT). If the record is found, the found flag is set to on, and the record is removed. If the record is not found, the flag is set to off. Then control is returned to the calling program.

Writing Programs to Query a User's Mailbox

The following program queries a specified mail box, and receives the mail into a specified file. The file is the output file layout of RCV DST.

Following are the OS/400 office commands used in this example:

QRYDST Queries a specified user's mail box, and writes the distribution records to an output file.

RCVDST Receives a distribution, and writes it to the file specified in the program.

Following is the program, RCVALLDST:

```
PGM          PARM(&USRID &ADDR &FILENAM &FILELIB)
/* This program will receive all the incoming      */
/* distributions sent to the user specified by &USRID */
/* and &ADDR and put into the file &FILENAM in library*/
/* &FILELIB.                                         */
/*                                                  */
/* The program does a QRYDST of the &USRID and &ADDR */
/* to an outfile. It processes each record in the   */
/* QRYDST outfile and does a RCV DST to receive each */
/* distribution.                                     */
/*                                                  */
/* For example if you want to receive all of the   */
/* distributions sent to user CHARLIE at address    */
/* CHICAGO and put the distributions in file RCV DST */
/* library OFCLIB, you would issue the following   */
/* command:                                         */
/* CALL RCVALLDST PARM(CHARLIE CHICAGO RCV DST OFCLIB) */
/*                                                  */
/* To create this program, use the CRTCLPGM command. */
/* This is the source code for the program.        */
/*                                                  */
DCL          VAR(&USRID) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADDR) TYPE(*CHAR) LEN(8)
DCL          VAR(&FILENAM) TYPE(*CHAR) LEN(10)
DCL          VAR(&FILELIB) TYPE(*CHAR) LEN(10)
DCLF        FILE(QSYS/QAOSILIN) RCDfmt(OSLIN) /* System +
QRYDST outfile layout */
```

Figure 4-20 (Part 1 of 2). RCVALLDST Program


```

/* Do a QRYDST to an outfile in library QTEMP          */
QRYDST      USRID(&USRID &ADDR) OUTFILE(QTEMP/QRYDSTIN)

/* Now override the system file with the file just    */
/* created by QRYDST.                                  */
OVRDBF      FILE(QAOSILIN) TOFILE(QTEMP/QRYDSTIN)

/* Read the records sequentially from the QRYDST      */
/* outfile.                                            */
READF:      RCVF      RCDFMT(OSLIN)

/* Monitor for EOF message                            */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
/* Receive the distribution and put into outfile that */
/* was input. Add the distribution to the file and    */
/* put all the records from the distributions to the  */
/* file. This also deletes the distribution.          */
RCVDST      DSTID(&LINDID) DSTIDEXN(&LINDEX) +
              USRID(&LINRUI &LINRUA) DOC(*NONE) +
              OUTFILE(&FILELIB/&FILENAM) OUTMBR(*FIRST +
              *ADD) OUTDTATYP(*ALL)

/* Read another record from the QRYDST outfile.      */
GOTO        CMDLBL(READF)

EOF:        /* Delete the temporary QRYDST outfile.    */
DLTF        FILE(QTEMP/QRYDSTIN)

ENDPGM

```

Figure 4-20 (Part 2 of 2). RCVALLDST Program

Converting a Spooled File to FFTDCA and Distributing the File

A command can integrate customer applications with OfficeVision/400. In this example, an accounts receivable application creates a printed report showing all overdue accounts. The report is five pages. Copies are printed and distributed to four managers who review the report and create action plans.

By using a combination of programming interfaces and customer programs, this same report can be distributed to the four managers so they see the report at their OfficeVision/400 work stations rather than on paper. If the managers were geographically dispersed, the solution could provide faster action plan development.

Following are two ways to deal with the spooled file:

- The spooled file can be copied into a database file using the Copy Spooled File (CPYSPF) command. A program can be written to convert the data stream to FFTDCA. The SNDDST command is used to distribute the data stream from a database file. A simple example of this approach follows.
- The RUN instruction supported with the PRTDOC command to embed the spooled file in a document. The printed version of the document can be saved in a folder as an FFTDCA document. Then, the SNDDST command is used to distribute the document as an FFTDCA document.

Add the following functions to an existing application to convert a spooled printer file to FFTDCA and distribute it.

```
CRTPF FILE(QTEMP/SPOOL) RCDLEN(136) SIZE(*NOMAX)
```

```
CRTPF FILE(QTEMP/FFTDCA) RCDLEN(133) SIZE(*NOMAX)
```

```
CPYSPLF FILE(NAME-OF-FILE) TOFILE(QTEMP/SPOOL) JOB(NAME-OF-JOB)
CTLCHAR(*PRTCTL)
```

```
CALL PGM(FFTDCA)
```

```
SNDDST TYPE(*FILE) DOCFILE(QTEMP/FFTDCA) DOCTYPE(*FFT)
TOUSERID(NAME-OF-RECIPIENTS) DSTD('FFT spooled document')
```

```
DLTF FILE(QTEMP/SPOOL)
```

```
DLTF FILE(QTEMP/FFTDCA)
```

The foregoing functions can be added as the following:

- Commands in an input stream
- A compiled CL program called at the end of the application
- A separate step that runs when the existing application ends

Following is the code for the program that calls the FFTDCA program (in CL):

```

/*****
/*
/* Program name: FFTDCA
/*
/* Descriptive name: Control conversion of spooled file to FFTDCA
/*
/* Author: IBM
/*
/* Function: This program is one of two programs used to convert
/* a file created by the Copy Spooled File (CPYSPLF)
/* command to a file using Final Form Text Document
/* Content Architecture (FFTDCA) data.
/*
/* For demonstration purposes, the main conversion
/* program was written in COBOL. However, COBOL does
/* not provide an easy method to declare hexadecimal
/* character strings.
/*
/* A number of FFTDCA controls are needed in order to
/* properly set up the format of the forms the data
/* will be printed on. These controls are hard coded
/* as hexadecimal character strings in this program and
/* passed to the COBOL program that does the bulk of
/* the conversion.
/*
/* Had another language been used instead of COBOL
/* (such as PL/I or PASCAL), this CL program would
/* not have been needed since those languages have
/* better support for declaring hex character strings.
/*

```

Figure 4-21 (Part 1 of 5). FFTDCA Program

```

/*          No attempt was made to design this conversion          */
/*          function as a general purpose function.  For example,  */
/*          it is assumed that all output conforms to the         */
/*          following requirements:                                */
/*          lines per inch = 8                                    */
/*          characters per inch = 10                             */
/*          form width = 14.5 inches                             */
/*          form depth = 8.5 inches                             */
/*          68 lines per form                                    */
/*          132 characters per line                              */
/*
/*          However, it would not be difficult to take the general */
/*          ideas presented in this program and the CONVERT program*/
/*          and generalize them in an implementation that would   */
/*          support different form sizes or lines per inch.      */
/*
/*          A call to this program in conjunction with several    */
/*          other CL commands can be used to convert a spooled   */
/*          printer file to FFTDCA and then either distribute that*/
/*          file in an IBM office network or file it in the      */
/*          document library.                                     */
/*
/*          For example:                                         */
/*
/*          -----
/*          | existing                                     |
/*          | application                                  |
/*          | runs and                                     |
/*          | produces a                                   |
/*          | spooled printer                             |
/*          | file                                         |
/*          |----->|                                     |
/*          |                                     |
/*          |                                     |
/*          |                                     |
/*          |                                     |
/*          |-----|                                     |
/*          | CL Program .                               |
/*          | .                                         |
/*          | CPYSPLF * * * * * >-----|
/*          |                                     |
/*          | CALL FFTDCA < . . . .| *PRTCTL Database |
/*          | *                                     | .format | file |
/*          |-----|                                     |
/*          | SNDDST *                                     |
/*          | - or -<. *                                     |
/*          | FILDOC . * * * * * * >| FFTDCA Database |
/*          |                                     | .format | file |
/*          |-----|                                     |
/*
/*          In order to change this program, FFTDCA controls must */
/*          be understood.  Refer to the following IBM publication */
/*          for information.
/*
/*          Document Interchange Architecture:  Final-Form
/*          Text Reference; GC23-0757
/*

```

Figure 4-21 (Part 2 of 5). FFTDCA Program

```

/*****/
PGM
/*****/
/* */
/* Variable used to combine all the FFTDCA controls that define the */
/* form and pass them as one parameter. */
/* */
/*****/
DCL VAR(&FORMDEF) TYPE(*CHAR) LEN(71)
/*****/
/* */
/* Set Exception Action */
/* Take action on all exception classes. */
/* Ignore exceptions. */
/* */
/*****/
DCL VAR(&SEA) TYPE(*CHAR) LEN(6) +
      VALUE(X'2BD204850001')

/*****/
/* */
/* Page Presentation Media */
/* Deactivate envelope selection and select paper. */
/* Source drawer 1. */
/* Use current destination drawer offset. */
/* Use current destination drawer. */
/* Use current print quality value. */
/* Use current duplex setting. */
/* */
/*****/
DCL VAR(&PPM) TYPE(*CHAR) LEN(12) +
      VALUE(X'2BD20A480000010100000000')

/*****/
/* */
/* Set Presentation Page Size */
/* Width = 20,880 1440ths of an inch (14.5 inches) */
/* Depth = 12,240 1440ths of an inch (8.5 inches) */
/* */
/*****/
DCL VAR(&SPPS) TYPE(*CHAR) LEN(8) +
      VALUE(X'2BD2064051902FD0')

/*****/
/* */
/* Set CGCSGID */
/* Graphic Character Set Global ID = 101 */
/* Code Page Global ID = 256 */
/* */
/*****/
DCL VAR(&SCG) TYPE(*CHAR) LEN(8) +
      VALUE(X'2BD1060100650100')

```

Figure 4-21 (Part 3 of 5). FFTDCA Program

```

/*****
/*
/* Set FID thru GFID
/* Global Font ID = 12 (Prestige Pica)
/* Font width = 144 1440ths of an inch (0.1 inches)
/* Font is monospaced
/*
/*****
DCL VAR(&SFG) TYPE(*CHAR) LEN(9) +
    VALUE(X'2BD10705000C009001')

/*****
/*
/* Set Single Line Distance
/* Line distance = 180 1440ths of an inch (1/8 of an inch or
/* 8 lines per inch)
/*
/*****
DCL VAR(&SSLD) TYPE(*CHAR) LEN(6) +
    VALUE(X'2BD2041500B4')

/*****
/*
/* Set Horizontal Margins
/* Left margin = left edge of presentation space
/* Right margin = 20,880 1440ths of an inch (14.5 inches)
/*
/*****
DCL VAR(&SHM) TYPE(*CHAR) LEN(8) +
    VALUE(X'2BD2061100015190')

/*****
/*
/* Set Line Spacing
/* Single line spacing
/*
/*****
DCL VAR(&SLS) TYPE(*CHAR) LEN(5) +
    VALUE(X'2BD2030902')

/*****
/*
/* Set Vertical Margins
/* Top margin = Use first vertical position at top of page
/* Bottom margin = Ignored
/*
/*****
DCL VAR(&SVM) TYPE(*CHAR) LEN(8) +
    VALUE(X'2BD2064900010000')

```

Figure 4-21 (Part 4 of 5). FFTDCA Program

```

/*****
/*
/* Form Feed
/*
/*****
DCL VAR(&FORMFEED) TYPE(*CHAR) LEN(1) VALUE(X'0C')
/*****
/*
/* Carriage Return
/*
/*****
DCL VAR(&CARGRETURN) TYPE(*CHAR) LEN(1) VALUE(X'0D')

/*****
/*
/* New Line
/*
/*****
DCL VAR(&NEWLINE) TYPE(*CHAR) LEN(1) VALUE(X'15')

/*****
/*
/* Concatenate the final form text controls together and pass them
/* to the spooled file conversion program as one parameter.
/*
/*****
CHGVAR VAR(&FORMDEF) VALUE(&SEA *CAT &PPM *CAT &SPPS *CAT &SCG +
      *CAT &SFG *CAT &SSLD *CAT &SHM *CAT &SLS *CAT &SVM)

/*****
/*
/* Call the conversion program to read the file created by the
/* Copy Spooled File (CPYSPLF) command and convert the records to
/* a final form text data stream.
/*
/*****
CALL PGM(CONVERT) PARM(&CARGRETURN &FORMFEED &NEWLINE &FORMDEF)

ENDPGM
/*****
/*
/* End of program FFTDCA
/*
/*****

```

Figure 4-21 (Part 5 of 5). FFTDCA Program

Following is the code for the COBOL program that converts the copied spooled file to FFTDCA.

```
PROCESS OPTIONS.
IDENTIFICATION DIVISION.
/*****/
/*
/* This program does a reasonably good job of converting
/* spooled printer files to an FFTDCA data stream. The
/* converted file can then be used as input to the File
/* Document (FILDOC) or Send Distribution (SNDDST) command.
/*
/* Because of some limitations with the Copy Spooled File
/* (CPYSPLF) command that is used to create the file that is
/* input to this program, some information is lost. For
/* example, a spooled printer file that contains graphics or
/* image data will lose the graphics and image data. Things
/* like underlining and highlighting get copied as multiple
/* print records. While this can be converted to FFTDCA, it
/* is not done in the normal FFTDCA fashion and thus may
/* not be displayed in OfficeVision/400 as would be expected.
/* OfficeVision/400 should still be able to print it
/* correctly.
/*
/* When this conversion program is used with typical data
/* processing application spooled printer files, the results
/* should be satisfactory. Typical data processing reports
/* produced by high-level languages such as RPG and COBOL do
/* not use functions that will pose a problem in the conversion.*/
/*
/* Of course, clever programmers can provide additional
/* function that this program does not have and thus provide
/* a better conversion to FFTDCA.
/*
/*****/
PROGRAM-ID. CONVERT.
AUTHOR. IBM.
INSTALLATION. IBM.
DATE-WRITTEN. APRIL, 1988.
DATE-COMPILED.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-AS400.
OBJECT-COMPUTER. IBM-AS400.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT SPOOL-FILE ASSIGN TO DATABASE-SPOOL
ORGANIZATION IS SEQUENTIAL.
SELECT FFTDCA-FILE ASSIGN TO DATABASE-FFTDCA
ORGANIZATION IS SEQUENTIAL.
DATA DIVISION.
FILE SECTION.
```

Figure 4-22 (Part 1 of 6). COBOL Program to Convert the Copied Spooled File to FFTDCA

```

/*****/
/*
/* SPOOL-FILE is the file that contains the output of the
/* Copy Spooled File (CPYSPLF) command. When that command is
/* run, it must specify CTLCHAR(*PRTCTL). Also, this
/* program assumes that the record length of the records in
/* SPOOL-FILE are 132 bytes. It would be possible to generalize
/* the design of this program so that it could handle a variety
/* of forms widths and thus record lengths.
/*
/* The first 4 bytes of the record are printer control
/* characters. Refer to the CPYSPLF command in the CL
/* Reference manual for further discussion.
/*
/* The external name of this file is SPOOL and can be in any
/* library in the library list.
/*
/*****/
FD SPOOL-FILE
RECORD CONTAINS 136 CHARACTERS
LABEL RECORDS ARE OMITTED
DATA RECORD IS SPOOL-REC.
01 SPOOL-REC.
03 SKIP-CHARS PIC X(3).
03 SKIP-CHAR-2 REDEFINES SKIP-CHARS PIC 999.
03 SPACE-CHAR PIC X(1).
03 SPOOL-LINE PIC X(132).
03 IN-TBL REDEFINES SPOOL-LINE.
05 IN-CHAR PIC X OCCURS 132 TIMES.

/*****/
/*
/* FFTDCA-FILE is the file that contains the output of the
/* conversion process. This file can then be used as input to
/* the File Document (FILDOC) or Send Distribution (SNDDST)
/* commands.
/*
/* The records in this file are built by inserting the
/* appropriate FFTDCA controls and then copying a print line
/* from SPOOL-FILE. The last record in this file may not be
/* completely filled with print data. Nulls (X'00') are used
/* to pad the last record.
/*
/* The external name of this file is FFTDCA and can be in any
/* library in the library list.
/*
/*****/
FD FFTDCA-FILE
RECORD CONTAINS 133 CHARACTERS
LABEL RECORDS ARE OMITTED
DATA RECORD IS FFTDCA-REC.
01 FFTDCA-REC.
03 FFTDCA-LINE PIC X(133).
03 OUT-TBL REDEFINES FFTDCA-LINE.
05 OUT-CHAR PIC X OCCURS 133 TIMES.

```

Figure 4-22 (Part 2 of 6). COBOL Program to Convert the Copied Spooled File to FFTDCA


```

WORKING-STORAGE SECTION.
77 I          PIC S999 COMP-4.
77 J          PIC S999 COMP-4.
77 LINE-CTR   PIC S999 COMP-4 VALUE 1.
77 IN-PTR     PIC S999 COMP-4 VALUE 1.
77 OUT-PTR    PIC S999 COMP-4 VALUE 1.
77 SPOOL-EOF  PIC X VALUE "0".
01 LOOP-CONTROL PIC 999.
01 LOOP-CONTROL-3 REDEFINES LOOP-CONTROL.
    03 FILLER PIC XX.
    03 LOOP-CONTROL-2 PIC X.

/*****/
/*
/* FFTDCA controls passed from the CL program.
/*
/*
/*****/
LINKAGE SECTION.
01 CARRIAGE-RETURN PIC X.
*
01 FORM-FEED PIC X.
*
01 NEW-LINE PIC X.
*
01 FFTDCA-SETUP.
    03 SEA PIC X(6).
    03 PPM PIC X(12).
    03 SPSS PIC X(8).
    03 SCG PIC X(8).
    03 SFG PIC X(9).
    03 SSLD PIC X(6).
    03 SHM PIC X(8).
    03 SLS PIC X(5).
    03 SVM PIC X(8).
PROCEDURE DIVISION USING CARRIAGE-RETURN FORM-FEED NEW-LINE
    FFTDCA-SETUP.

MAIN-BODY.
    PERFORM INIT.

/*****/
/*
/* Process all records in the file created by CPYSPLF.
/*
/*
/*****/
    PERFORM PROC-SPOOL-FILE UNTIL SPOOL-EOF = "1".
    PERFORM CLEAN-UP.
    STOP RUN.

INIT.
    OPEN INPUT SPOOL-FILE.
    OPEN OUTPUT FFTDCA-FILE.

```

Figure 4-22 (Part 3 of 6). COBOL Program to Convert the Copied Spooled File to FFTDCA

```

/*****/
/* Move the definition of the form and other FFTDCA controls to */
/* the first record. */
/*****/
      MOVE FFTDCA-SETUP TO FFTDCA-REC.
      MOVE 71 TO OUT-PTR.

PROC-SPOOL-FILE.
  PERFORM READ-RECORD.
/*****/
/* Test to see if this record has a space control character or */
/* a skip control character. */
/*****/
      IF SPOOL-EOF NOT EQUAL "1" THEN
        IF SKIP-CHARS NOT EQUAL " " THEN
          PERFORM PROC-SKIP
        ELSE
          PERFORM PROC-SPACE
      ELSE
        NEXT SENTENCE.
/*****/
/*
*/
/* Reduce the amount of data by trimming trailing blanks from */
/* each print line. */
/*
*/
/*****/
      IF SPOOL-EOF NOT EQUAL "1" THEN
        IF SPOOL-LINE EQUAL ALL SPACES THEN
          MOVE 0 TO J
        ELSE
          PERFORM TRIM-BLANKS
            VARYING J FROM 132 BY -1
            UNTIL IN-CHAR(J) NOT EQUAL " "
      ELSE
        NEXT SENTENCE.
/*****/
/* Copy the print data from the input record to the output */
/* record. */
/*****/
      IF SPOOL-EOF NOT EQUAL "1" THEN
        PERFORM COPY-DATA
          VARYING IN-PTR FROM 1 BY 1
          UNTIL IN-PTR GREATER THAN J
      ELSE
        NEXT SENTENCE.

READ-RECORD.
  READ SPOOL-FILE RECORD AT END MOVE "1" TO SPOOL-EOF.

PROC-SKIP.

```

Figure 4-22 (Part 4 of 6). COBOL Program to Convert the Copied Spooled File to FFTDCA

```

/*****/
/*
/* If we are skipping to a line number that is less than the
/* current line number, we will cross a page boundary so insert
/* a form feed. For example if we are currently on line 10 and
/* encounter a skip to line 1, we are skipping to line 1 of the
/* next page and need to insert a form feed first.
/*
/*
/*****/
      IF SKIP-CHAR-2 LESS THAN LINE-CTR THEN
        PERFORM INSERT-FORM-FEED
      ELSE
        NEXT SENTENCE.
      SUBTRACT LINE-CTR FROM SKIP-CHAR-2 GIVING LOOP-CONTROL.
/*****/
/*
/* Use multiple new line characters to implement a skip.
/*
/*
/*****/
      PERFORM INSERT-NEW-LINE
        VARYING I FROM 1 BY 1
        UNTIL I GREATER THAN LOOP-CONTROL.

PROC-SPACE.
  MOVE 0 TO LOOP-CONTROL.
  MOVE SPACE-CHAR TO LOOP-CONTROL-2.
  IF LOOP-CONTROL GREATER THAN 0 THEN
    PERFORM INSERT-NEW-LINE
      VARYING I FROM 1 BY 1
      UNTIL I GREATER THAN LOOP-CONTROL
  ELSE
    PERFORM INSERT-CARRIAGE-RETURN.

INSERT-FORM-FEED.
  MOVE 1 TO LINE-CTR.
  MOVE FORM-FEED TO OUT-CHAR(OUT-PTR).
  ADD 1 TO OUT-PTR.
  IF OUT-PTR GREATER THAN 133 THEN
    PERFORM WRITE-RECORD
  ELSE
    NEXT SENTENCE.

INSERT-NEW-LINE.
  MOVE NEW-LINE TO OUT-CHAR(OUT-PTR).
  ADD 1 TO OUT-PTR.
  IF OUT-PTR GREATER THAN 133 THEN
    PERFORM WRITE-RECORD
  ELSE
    NEXT SENTENCE.
  ADD 1 TO LINE-CTR.

```

Figure 4-22 (Part 5 of 6). COBOL Program to Convert the Copied Spooled File to FFTDCA

```

INSERT-CARRIAGE-RETURN.
    MOVE CARRIAGE-RETURN TO OUT-CHAR(OUT-PTR).
    ADD 1 TO OUT-PTR.
    IF OUT-PTR GREATER THAN 133 THEN
        PERFORM WRITE-RECORD
    ELSE
        NEXT SENTENCE.

TRIM-BLANKS.
    EXIT.

COPY-DATA.
/*****/
/*
/* Copy one print character from the input record to the output */
/* record. If the output record has been filled after the */
/* copy, write it to the file and start over with a new record. */
/*
/*
/*****/
    MOVE IN-CHAR(IN-PTR) TO OUT-CHAR(OUT-PTR).
    ADD 1 TO OUT-PTR.
    IF OUT-PTR GREATER THAN 133 THEN
        PERFORM WRITE-RECORD
    ELSE
        NEXT SENTENCE.

WRITE-RECORD.
    WRITE FFTDCA-REC.
    MOVE 1 TO OUT-PTR.
/*****/
/* Initialize the output record with nulls (X'00') so the last */
/* record is properly padded if the last record is not filled */
/* with print data. */
/*
/*****/
    MOVE ALL LOW-VALUES TO FFTDCA-REC.

CLEAN-UP.
/*****/
/* Test to see if the last record has at least one printable */
/* character in it. If it does, write it to the output file. */
/*
/*****/
    IF OUT-PTR GREATER THAN 1 THEN
        PERFORM WRITE-RECORD
    ELSE
        NEXT SENTENCE.
    CLOSE SPOOL-FILE FFTDCA-FILE.
/*****/
/* End of program CONVERT */
/*****/

```

Figure 4-22 (Part 6 of 6). COBOL Program to Convert the Copied Spooled File to FFTDCA

Maintaining Mail Logs

The example in this section illustrates the use of commands in system management functions. In this application the commands, embedded in CL programs, help manage the amount of storage used by incoming mail. The application checks the incoming mail of system users, counts the number of distribution items each user has, sends a warning to the user the first time the items exceed a specified threshold, and deletes mail if the user does not clean up his mail.

This approach does not work for maintaining the outgoing mail if the mail was sent from the OfficeVision/400 product. Outgoing mail in this case needs to be maintained interactively through the OfficeVision/400 product, or by using the DLTOUTMAIL command explained in “Delete Outgoing Mail (DLTOUTMAIL) Command” on page 10-3. The DLTOUTMAIL command only deletes outgoing mail for distributions in which confirmation of delivery was not requested. For mail in which confirmation of delivery was requested, use the Delete Distribution (DLTDST) command. See the sample program in “Deleting Outgoing Mail” on page 4-96.

Following are the OS/400 office commands used in this application:

| | |
|---------------|--|
| DSPDIR | The Display Directory command creates a database file with one record for each user in the system directory. |
| QRYDST | The Query Distribution command sends a record to a database file for each incoming distribution for a user. |
| SNDDST | The Send Distribution command is used to send messages to the users. |
| DLTDST | The Delete Distribution command removes items from the incoming distributions for a user. |

Assumptions

It is assumed that all users of document distribution services are enrolled in the system directory. The directory is used as a guide to query incoming mail. If a user is not enrolled in the directory, no check is made.

The person calling the INMAIL program to start the application should have authority to handle mail on behalf of all users in the system directory. Otherwise, the application ends in error.

It is assumed that user IDs are unique, and that all users are local users. Unique user IDs are assumed because even when multiple ID entries exist in the system directory for the same user, they represent one user and one incoming mail bucket. The application runs successfully even when duplicate IDs exist. The assumption is made that all users are local because remote users should not have mail sent to this system. It should be made to their local system. If the application tries to query incoming mail for a remote user, unpredictable results can occur. Therefore, add a program to this application to go through the output file of the DSPDIR command and sort duplicate IDs and remote users.

The application uses four database files:

| | |
|---------------|---|
| USRLST | The output of the DSPDIR command that contains a list of the users in the system directory. |
|---------------|---|

- MAIL** The output for the QRYDST command that contains the mail items for each user.
- OVERLMT** Contains the names of users with too many incoming mail items.
- FGND** Contains the names of users to be deleted that are not found.

Following is the record layout of the USRLST file:

```

A** DDS OBJECT:      USRLST
A** SOURCE FILE:    QDSSRC
A** SOURCE LIB:     OFCLIB
A** DESCRIPTION:    DISTRIBUTION LIST TO BE USED IN 'QRYDST'
A*
A      R OSDIRE
A      USR          8A
A      ADD          8A
A      DSC          50A
A      SYSN         8A
A      SYSG         8A
A      USRP         10A
A      INDF         1A
A      PERF         1A
A      ADDR1        36A
A      ADDR2        36A
A      ADDR3        36A
A      ADDR4        36A
A      LOC          36A
A      TEL1         25A
A      TEL2         25A
A      TXT          50A
A      K USR

```

Note: Because this file is the output for the DSPDIR command, the record format name (OSDIRE) in USRLST must be the same as that specified by the command. Otherwise, the command ends in error. This may hinder customers who have naming conventions for files and record formats. See Appendix C, "Office Services Output File Considerations," for information about the fields.

Following is the record format for the file, MAIL:

```
A** DDS OBJECT:      MAIL
A** SOURCE FILE:     QDDSSRC
A** SOURCE LIB:      OFCLIB
A** DESCRIPTION:     OUTPUT FILE FOR QRYDST COMMAND
A*
A          R OSLIN
A          LINSNO      6S
A          LINDID      20A
A          LINRUI      8A
A          LINRUA      8A
A          LINSUI      8A
A          LINSUA      8A
A          LINSDT      8A
A          LINSTM      6A
A          LINPER      1A
A          LINPTY      1A
A          LINCOD      1A
A          LINDTP      5S
A          LINSYC      13A
A          LINCID      3S
A          LINCPG      3S
A          LINDSD      44A
A          LINDEX      2S
A          LINMLS      1A
A          LINC DT      8A
A          LINCTM      6A
A          K LINSDT
```

This example file is the output of the command QRYDST. Each record in the file represents a distribution item. The file represents the incoming mail for the user when the command runs.

The record format name (OSLIN) must be the same as specified by the QRYDST command. See Appendix C, "Office Services Output File Considerations," for a description of the fields for the QRYDST command.

The following example is the record layout for the file, OVERLMT:

```
A** DDS OBJECT:      OVERLMT
A** SOURCE FILE:     QDDSSRC
A** SOURCE LIB:      OFCLIB
A** DESCRIPTION:     USERS WITH TOO MUCH INCOMING MAIL
A*
A          R OVERLMTRF
A          GUSR        8A
A          GADD        8A
A          GDSC        50A
A          GSTAMP      3 0
A          K GUSR
```

This file contains the names of the users who have too many incoming mail items. The GSTAMP field represents the date the record was added to the file (the date the user was found to have too many mail items). This field is used by the MLCHK2 program to determine whether the 7-day grace period is over for the user.

Following is the record format for the FGND file:

```
A** DDS OBJECT:      FGND
A** SOURCE FILE:    QDSSRC
A** SOURCE LIB:     JOSEL
A** DESCRIPTION:    DELINQUENT USERS TO BE DELETED BUT NOT FOUND
A*
A          R GND
A          USR          8A
A          ADD          8A
A          DSC          50A
A          STAMP        6 0
A          K USR
```

Code

This application consists of three CL programs and two RPG programs. The CL program INMAIL is the main program. INMAIL runs the DSPDIR command to an output file to get a list of all users. Then the output file is read one record at a time. Each record represents a user on the system.

Incoming mail for each user is queried. The QRYDST command is used with an output file option to get a list in a database file of the distribution items for the user. Each record in the file represents a mail item. Then INMAIL receives the completion message from QRYDST and calls the RPG program, CVTBINDEC, to convert the message data to decimal and determine the number of mail items in the output file.

Then INMAIL compares the number of mail items to the threshold. For this application, the threshold is 15 items. However, you can pass the threshold as a parameter when the INMAIL program is called. If the number of items is less than the threshold, the program checks whether the user name should be deleted from the file with the names of the delinquent users. If the name should be deleted, INMAIL calls the RPG program, REMOVE, passing to it the user name as a parameter. REMOVE deletes the record representing the user from the delinquent file, and returns control to INMAIL. Then, the program branches to get the next user.

If the number of items is above the threshold, the RPG program MLCHK2 is called. The name and address of the user with too many items, and the current date, are passed to the program.

MLCHK2 determines whether the user is a first time offender. If this is the first time the user's mail items have exceeded the threshold, MLCHK2 adds his name to the file of delinquent users (OVERLMT), and returns control to INMAIL indicating a message should be sent to the user telling him that he has 7 days to clean up his mail (7 days is a number you can choose that is best passed as a parameter in the initial call to INMAIL).

However, if the user is delinquent again, MLCHK2 determines when the first warning was sent to the user by comparing the time stamp in OVERLMT with the current date. If the user is within the 7-day grace period, the name remains in the delinquent file. Control is returned to INMAIL with the number of days left in the grace period. Then another message can be sent to the user. If the user has been warned before the grace period has elapsed, and the user is still delinquent,

MLCHK2 determines the deletion date (the mail older than the specified date will be deleted).

In this example, the deletion date is 30 days prior to the current date. This is another parameter best determined by the person responsible for the application. It can be passed to INMAIL as a parameter. After the deletion date is set, control is returned to INMAIL with the deletion date and a flag to indicate to INMAIL that the user's mail needs to be cleaned up.

When INMAIL receives control from MLCHK2, it determines whether mail needs to be deleted or a message needs to be sent to the user. The delete flag passed by MLCHK2 is checked. A message is sent if the flag is negative. If the flag is positive, INMAIL calls the CL program DLTML to perform the mail deletion.

DLTML compares the deletion date with the stop deleting date. The stop deleting date is provided so recent mail is not deleted. In this example, the delete date is 14 days prior to the current date so mail less than 2 weeks old will not be deleted. If the deletion date is within the stop deleting date (within the 2-week grace period), mail is not deleted even if the count is above the threshold. Instead, a message is sent telling the user to clean up his mail. Control is passed to INMAIL where processing continues with the next user.

If the deletion date is outside the stop deleting date, the cleanup process begins. Each mail item for the user is read, and its date is compared with the deletion date. If the item is older than the deletion date, it is deleted. The count is decreased by one and compared with the threshold. If the threshold is still exceeded, the next item is read and the process is repeated. This process continues until the count is reduced to the threshold or the end of the file is reached.

If enough mail items are deleted to bring the count under the threshold, DLTML sends a message informing the user of the number of items deleted and the deletion date. Control is returned to INMAIL and processing continues with the next user. However, if the end of the file is reached, and the count is still above the threshold, DLTML sets a flag to let INMAIL know that a new deletion date closer to the current date is needed. Control is passed to INMAIL, and DLTML waits for the next deletion date.

When returning from DLTML, INMAIL decides whether the current user needs a new deletion date or processing should continue. If processing should continue, a branch is done to read the next record in the user's file. If a new deletion date is required for the current user, control is passed to MLCHK2 where the new deletion date is set to 7 days after the last deletion date. The new date is passed to DLTML and the process repeats until the count is below the threshold, or the deletion date gets within the stop deleting date.

Following is the code for the main program, INMAIL:

```
PGM
DCLF      FILE(OFCLIB/USRLST)
DCL       VAR(&CNT) TYPE(*DEC) LEN(9 0)
DCL       VAR(&FRST) TYPE(*CHAR) LEN(1) VALUE('Y')
DCL       VAR(&DLT) TYPE(*CHAR) LEN(1) VALUE('N')
DCL       VAR(&NXTUSR) TYPE(*CHAR) LEN(1) VALUE('N')
DCL       VAR(&SYSDAT) TYPE(*DEC) LEN(3 0)
DCL       VAR(&DLTDAT) TYPE(*DEC) LEN(3 0)
DCL       VAR(&STPDAT) TYPE(*DEC) LEN(3 0)
DCL       VAR(&JULDAT) TYPE(*CHAR) LEN(5)
DCL       VAR(&JULDDD) TYPE(*CHAR) LEN(3)
DCL       VAR(&ADATE) TYPE(*CHAR) LEN(6)
DCL       VAR(&AFMT) TYPE(*CHAR) LEN(4)
DCL       VAR(&FMT) TYPE(*CHAR) LEN(3)
DCL       VAR(&DAYS) TYPE(*DEC) LEN(3 0)
DCL       VAR(&DAYSC) TYPE(*CHAR) LEN(4)
DCL       VAR(&MSG1) TYPE(*CHAR) LEN(70)
DCL       VAR(&MSGID) TYPE(*CHAR) LEN(7)
DCL       VAR(&MSGDTA) TYPE(*CHAR) LEN(4)
DCL       VAR(&YEAR) TYPE(*CHAR) LEN(2)
DSPDIR    OUTPUT(*OUTFILE) OUTFILE(OFCLIB/USRLST) +
          DETAIL(*FULL) /* Create the list of users +
          in the system to be queried. */
RTVSYSVAL SYSVAL(QDATE) RTNVAR(&ADATE) /* Retrieve +
          system current date. */
RTVSYSVAL SYSVAL(QDATFMT) RTNVAR(&FMT)
CHGVAR    VAR(&AFMT) VALUE('*' *CAT &FMT)
CVTDAT    DATE(&ADATE) TOVAR(&JULDAT) FROMFMT(&AFMT) +
          TOFMT(*JUL) TOSEP(*NONE) /* Convert the +
          system date to Julian date. */
CHGVAR    VAR(&YEAR) VALUE(*%SST(&JULDAT 1 2))
CHGVAR    VAR(&JULDDD) VALUE(*%SST(&JULDAT 3 3))
CHGVAR    VAR(&SYSDAT) VALUE(&JULDDD) /* Convert from +
          Julian character format to Julian decimal. */
CHGVAR    VAR(&STPDAT) VALUE(&SYSDAT - 14) /*SET STOP +
          DATE. */
```

Figure 4-23 (Part 1 of 2). INMAIL Program

```

NXTRCD:   RCVF      RCD_FMT(OSDIRE) /* Read the names of the users +
          one at a time until the end of file is +
          reached.                                     */
MONMSG   MSGID(CPF0864) EXEC(GOTO CMDLBL(END))
QRYDST   USRID(&USR &ADD) OUTFILE(OFCLIB/MAIL) /* +
          Query the incoming distributions for each +
          user; for each item of mail, one record is +
          created in file MAIL.                       */
CHGVAR   VAR(&MSGID) VALUE('CPC9030')
RCVMSG   MSGID(&MSGID) MSGTYPE(*COMP) MSGDTA(&MSGDTA)
CALL     PGM(OFCLIB/CVTBIN4DEC) PARM(&MSGDTA &CNT)
SECOND:  IF        COND(&CNT *LT 15) THEN( +
          DO)
          IF COND(&FRST *EQ 'Y') THEN(GOTO +
          CMDLBL(NXTRCD))
          ELSE +
          DO
          CALL PGM(REMOVE) PARM(&USR)
          CHGVAR VAR(&FRST) VALUE('Y')
          GOTO CMDLBL(NXTRCD)
          ENDDO
          ENDDO
ELSE +
DO
CALL     PGM(MLCHK2) PARM(&USR &ADD &DSC &SYSDAT +
          &FRST &DLT &DLTDAT &DAYS)
IF COND(&DLT *EQ 'N') THEN( +
DO)
CHGVAR VAR(&DAYSC) VALUE(&DAYS)
CHGVAR VAR(&MSG1) VALUE('**** YOU HAVE ' +
*CAT &DAYSC *CAT ' DAYS TO CLEAR YOUR MAIL****')
SNDDST TYPE(*MSG) TOUSRID((&USR &ADD)) +
DSTD('SEND WARNING MESSAGE') MSG(&MSG1)
GOTO CMDLBL(NXTRCD)
ENDDO
ELSE +
DO
CALL PGM(OFCLIB/DLTM) PARM(&USR &ADD &DLTDAT +
&STPDAT &FRST &DLT &CNT &NXTUSR &YEAR) /* CALL+
THE PROGRAM WHICH WILL DELETE THE MAIL ITEMS+
OLDER THAN THE DELETE DATE */
IF COND(&NXTUSR *EQ 'Y') THEN(GOTO CMDLBL(NXTRCD))
GOTO CMDLBL(SECOND)
ENDDO
ENDDO
END:     ENDPGM

```

Figure 4-23 (Part 2 of 2). INMAIL Program

The file USRLST is declared to the program for processing. The process cycle is repeated for each user, so the file is read one record at a time (one user at a time), and the appropriate processing is done for each user. The program variables are then declared. The DSPDIR command is run to create the list of users to the file USRLST. With the list, the process can begin.

Since arithmetic calculations are done on the system date or current date, it is converted to Julian and decimal format. Simultaneously, the stop deleting date is set to current date, minus 14. Then the first record is read. The user is the current

user until the program branches back to the RCVF statement and a new user is read in. The QRYDST command runs for the current user. The output of the command is routed to the file MAIL. The file now has a record for every item of incoming distributions for this user.

Then the message data from the QRYDST completion message is converted from character to decimal. INMAIL performs an IF statement to compare the count to the predetermined threshold. If the count is under the threshold the program then determines whether the user must be removed from the OVERLMT file. The flag &FRST indicates whether this is the first time around the cycle for this user (this flag is initially set to Y for each user and is reset to N in the MLCHK2 program if the user is found to be delinquent).

If the flag is Y, no more processing is required for this user and the program branches to the RCVF statement. If the flag is N, the REMOVE program is called to delete the user from the OVERLMT file.

If the number of items is above the threshold, the program branches to the ELSE DO statement. The program MLCHK2 is called to determine whether mail must be deleted for the user or a warning sent. MLCHK2 returns a flag (&DLT), the deletion date (&DLTDAT), or the number of days the user has before automatic deletion takes place.

When return from MLCHK2, the flag &DLT is checked. If it is negative, indicating that deletion should not be done, a message is sent to the user and the program branches to the RCVF statement.

If the &DLT flag is positive, INMAIL calls the program DLTML to perform the deletion of mail for the user. After the mail has been deleted, control is passed back to INMAIL where a decision is made either to go to the next user or to do additional processing for the current user.

If additional processing is done, the program branches back to the label SECOND. Processing for each user stops when one of the following conditions occurs:

- The count is under the threshold.
- The deletion date is within the stop deleting date.
- The user is within the 7-day grace period and a message is sent.

Following is the code for the RPG program, CVTBIN4DEC:

```

H** RPG OBJECT:          CVTBIN4DEC
H** SOURCE FILE:        QRPGRSRC
H** SOURCE LIB:         OFCLIB
H**
H** DESCRIPTION: CONVERT BINARY CHARACTER STRING TO DECIMAL VALUE
H*****
I*****
IDS1          DS
I              B    1    40VAR1B
I*
I*
C*
C          *ENTRY  PLIST
C              PARM          DS1    4
C              PARM VAR1B    VAR2    90
C              SETON                      LR
C              RETRN

```

CVTBIN4DEC is called by the main program, INMAIL, to convert the message data in character format (binary) to decimal format.

Following is the code for the RPG program, MLCHK2:

```

H** RPG OBJECT:          MLCHK2
H** SOURCE FILE:        QRPGRSRC
H** SOURCE LIB:         OFCLIB
H**
H** DESCRIPTION: THIS PROGRAM DETERMINES WHETHER MAIL ITEMS
H**              ARE TO BE DELETED, OR JUST A MESSAGE SENT TO
H**              THE USER TO CLEAN UP THE MAIL
H**
H** ABNORMAL
H** TERMINATION: RE-RUN THE CL PROGRAM DRIVING THIS PROGRAM
H**              TO CLEAR THE OUTPUT FILES AND TRY AGAIN
H**
H*****
H**              SUBROUTINE SUMMARY
H*****
H** *PSSR - STANDARD ERROR HANDLING ROUTINE
H*****
H**              INDICATOR USE
H*****
H**    10 - EOF NEW
H**    11 - NOT FOUND IN 'OVERLMT'
H*****
H** UPDATE FORMAT FOR THIS PGM YYMMDD - DEBUG=1 FOR DUMP CAPAB
H*****
H          1  Y

```

Figure 4-24 (Part 1 of 3). MLCHK2 RPG Program

```

F*****
F* FILE WITH DELINQUENT ENTRIES
F*****
FOVERLMT IF E          K          DISK          A
F                                KINFSR *PSSR
E*****
I*****
I* STANDARD INFORMATION DATA STRUCTURE - SYSTEM SUPPLIED
I*****
IERRDS      SDS
I                                *STATUS ESTS1
I                                *ROUTINE ROUT1
I                                *PARMS  PARM1
I                                *PROGRAM PGMNAM
I                                81  90 PGMLIB
I                                244 253 JOBNAM
I                                254 263 USRNAM
I                                264 2690JOBNUM
I                                282 2870JOBTIM
I*
I*
C*
C*
C*
C*****
C* DETERMINE WHETHER MAIL NEEDS TO BE DELETED
C*****
C*
C          *ENTRY  PLIST
C                                PARM          GUSR      8
C                                PARM          GADD      8
C                                PARM          GDSC      50
C                                PARM          DATE      30
C                                PARM          FRST       1
C                                PARM          DLTF LG    1
C                                PARM          DLT DAT    30
C                                PARM          DAYS      30
C*
C          FRST    IFEQ 'Y'
C*
C          GUSR    CHAINOVERLMTRF          11    NOT FOUND

```

Figure 4-24 (Part 2 of 3). MLCHK2 RPG Program

```

C*
C      *IN11      IFEQ '1'
C                MOVE DATE      GSTAMP
C                WRITEOVERLMTRF
C                ADD 1          TOTADD 70      ADD RCD
C                Z-ADD7        DAYS          ADDED RCD
C                MOVE 'N'      DLTFLG
C*
C                ELSE
C      DATE      SUB GSTAMP    DIFF    60
C      DIFF      IFGE 7
C                MOVE 'Y'      DLTFLG
C      DATE      SUB 30       DLTDAT
C*
C                ELSE
C      7         SUB DIFF      DAYS
C                MOVE 'N'      DLTFLG
C                END
C*
C                END
C*
C                ELSE
C                ADD 7         DLTDAT
C                MOVE 'Y'      DLTFLG
C                END
C                RETRN
C*
C*
C                MOVE '1'      *INLR
C                RETRN
C*
C*
C*****
C* PSSR          * - PROGRAM ERROR SUBROUTINE
C*****
C* TO DUMP DEBUG MUST BE EQUAL TO 1 (SEE H SPEC)
C*****
C      *PSSR     BEGSR
C      PGMNAM    DUMP
C                MOVE '1'      *INLR
C                RETRN
C                ENDSR
C*****
O*****

```

Figure 4-24 (Part 3 of 3). MLCHK2 RPG Program

MLCHK2 is called by the main program, INMAIL. In the file description specifications, the file OVERLMT is defined as input/output, externally defined, keyed, and residing in the disk.

In the calculation specifications, processing begins with the *ENTRY PLIST statement. The parameters passed between the two programs are defined. Then the flag FRST is checked to determine whether this is the first time around for this user. If it is not the first time, only a new deletion date is needed, and the program branches to get the date.

However, if the flag is Y, a search is done on the OVERLMT file (CHAIN statement) for the user. If the user is not found then the record is added (WRITE) to OVERLMT. If the user is found, the user is delinquent. The number of days since the first warning was sent is calculated by subtracting the date stamp from the current date. If the 7-day grace period is ended, the delete flag is set to ON, and the deletion date is set to the current date, minus 30 days. If the grace period has not ended, the number of days left before it ends is calculated, it is placed in the variable DAYS, and the delete flag is set to OFF. Control is returned to INMAIL with the delete flag, deletion date, and days parameters.

Following is the code for the CL program, DLTML:

```

/* THIS PROGRAM DELETES MAIL ITEMS FROM THE INCOMING MAIL FOR THE */
/* USER "&USR &ADD" WITH DATE STAMP OLDER THAN THE DELETE DATE */
/* "&DLTDAT" */
PGM          PARM(&USR &ADD &DLTDAT &STPDAT &FRST &DLT +
                &CNT &NXTUSR &YEAR)
/* DECLARE THE FILES AND VARIABLES USED BY THE PROGRAM */
DCLF         FILE(OFLIB/MAIL) /* File MAIL +
                contains the records resulting from the +
                QRYDST. It has a record format name OSLIN. */
DCL          VAR(&USR) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADD) TYPE(*CHAR) LEN(8)
DCL          VAR(&DLTDAT) TYPE(*DEC) LEN(3 0)
DCL          VAR(&STPDAT) TYPE(*DEC) LEN(3 0)
DCL          VAR(&DDATE) TYPE(*CHAR) LEN(8) /* The delete +
                date (DLTDAT) variable will be converted +
                from decimal to character and placed in +
                variable &DATE. */
DCL          VAR(&FRST) TYPE(*CHAR) LEN(1)
DCL          VAR(&DLT) TYPE(*CHAR) LEN(1)
DCL          VAR(&NXTUSR) TYPE(*CHAR) LEN(1)
DCL          VAR(&CNT) TYPE(*DEC) LEN(9 0)
DCL          VAR(&CNTLOC) TYPE(*DEC) LEN(9 0)
DCL          VAR(&CNTDIF) TYPE(*DEC) LEN(9 0)
DCL          VAR(&CNTDIFC) TYPE(*CHAR) LEN(9)
DCL          VAR(&MSG2) TYPE(*CHAR) LEN(100)
DCL          VAR(&DATE1) TYPE(*CHAR) LEN(3)
DCL          VAR(&DATE2) TYPE(*CHAR) LEN(5)
DCL          VAR(&DATE3) TYPE(*CHAR) LEN(6)
DCL          VAR(&YEAR) TYPE(*CHAR) LEN(2)
/* CONVERT THE &DLTDAT TO CHARACTER */
/*
CHGVAR      VAR(&DATE1) VALUE(&DLTDAT) /* Go from decimal +
                to character. */
CHGVAR      VAR(&DATE2) VALUE(&YEAR *tcat &date1)
CVTDAT      DATE(&DATE2) TOVAR(&DATE3) FROMFMT(*JUL) +
                TOFMT(*YMD) TOSEP(*NONE) /* Convert the +
                date from Julian format to Year-Month-Day +
                format. */
CHGVAR      VAR(&DDATE) VALUE('19' *cat &date3) /* +
                Convert to YYYYMMDD format. */

```

Figure 4-25 (Part 1 of 3). DLTML CL Program


```

/* SET THE LOCAL COUNT TO THE INITIAL COUNT PASSED IN          */
      CHGVAR      VAR(&CNTLOC) VALUE(&CNT)
/* COMPARE THE STOP-DELETING DATE TO THE DELETE DATE. IF THE DELETE*/
/* DATE IS WITHIN THE STOP-DATE RANGE EXIT THIS ROUTINE AND GO TO */
/* NEXT USER.                                                    */
/* EACH TIME DLTML IS RE-CALLED FOR THE SAME ARE ADDED TO &DLTDATE */
      IF COND(&DLTDATE *GT &STPDATE) THEN( +
DO)
      SNDDST TYPE(*MSG) TOUSRID((&USR &ADD)) +
      DSTD('WARNING') MSG('**** COULD YOU PLEASE REMOVE +
      SOME OF YOUR INCOMING MAIL? ****')
      CHGVAR VAR(&FRST) VALUE('Y')
      CHGVAR VAR(&DLT) VALUE('N')
      CHGVAR VAR(&NXTUSR) VALUE('Y')
      GOTO CMDLBL(OUT)
      ENDDO
      ELSE +
      DO
/*
/* CHECK THE COUNT NUMBER; IF COUNT IS LESS THAN THE THRESHOLD THEN*/
/* EXIT THIS ROUTINE AND GO TO NEXT USER.                        */
/*
START:      IF COND(&CNT *LT 15) THEN( +
DO)
      CHGVAR VAR(&CNTDIF) VALUE(&CNTLOC - &CNT)
      CHGVAR VAR(&CNTDIFC) VALUE(&CNTDIF)
      CHGVAR      VAR(&MSG2) VALUE('**** ' *CAT &CNTDIFC *CAT ' +
      MAIL ITEMS RECEIVED BEFORE ' *CAT &DDATE +
      *CAT ' WERE DELETED FROM YOUR INCOMING MAIL +
      ****')
      SNDDST TYPE(*MSG) TOUSRID((&USR &ADD)) +
      DSTD('DELETION MSG') MSG(&MSG2)
      CHGVAR VAR(&FRST) VALUE('N')
      CHGVAR VAR(&NXTUSR) VALUE('N')
      GOTO CMDLBL(OUT)
      ENDDO

```

Figure 4-25 (Part 2 of 3). DLTML CL Program

```

/*                                                                    */
/* THE COUNT IS ABOVE THE THRESHOLD SO BEGIN TO READ RECORDS AND    */
/* COMPARE THE DATE STAMP AGAINST THE DELETE DATE. IF THE TIME STAMP*/
/* IS OLDER THAN THE DELETE DATE THEN DELETE THE ITEM FROM THE     */
/* INCOMING MAIL. OTHERWISE READ THE NEXT RECORD AND REPEAT.      */
/*                                                                    */
ELSE +
DO
    RCVF RCD_FMT(OSLIN)
    MONMSG MSGID(CPF0864) EXEC(GOTO CMDLBL(SET))
    IF COND(&LINSDT *LT &DDATE) THEN( +
        DO)
        DLTST DSTID(&LINDID) OPTION(*IN) +
        USRID(&USR &ADD) DSTIDEXN(&LINDEX)
        CHGVAR VAR(&CNT) VALUE(&CNT - 1)
    ENDDO
    GOTO CMDLBL(START)
ENDDO
/* END OF IF*/
ENDDO

/*                                                                    */
/* SET THE FLAGS TO BRANCH APPROPRIATELY IN THE MAIN PROGRAM      */
/*                                                                    */
SET:      CHGVAR      VAR(&NXTUSR) VALUE('N')
          CHGVAR      VAR(&FRST)  VALUE('N')

OUT:      RETURN

ENDPGM

```

Figure 4-25 (Part 3 of 3). DLTML CL Program

INMAIL calls DLTML to delete mail from a user's incoming mail queue when delinquent. DLTML uses the &USR parameter and the MAIL file to determine the mail to be deleted. File MAIL and the parameters are declared for processing.

The deletion date passed by INMAIL is converted from decimal Julian to character YMD for comparison to the date sent stamp on each mail item. The deletion date is compared with the stop deleting date to ensure it is not within the 14-day buffer period. If the deletion date is within 14 days of the current date, nothing is deleted (even if the count is above the threshold), a message is sent to the user (to clean up the mail), the next user flag (&NXTUSR) is set to YES, and the program branches to the end.

If the deletion date is outside the stop deleting date, the count is checked against the threshold. If the count is above the threshold, the program deletes mail items. It compares the sent date of each item with the deletion date. The program deletes the items with a sent date older than the deletion date, decreases the count by one, and goes back to compare the count with the threshold. The process continues until the count is below the threshold, or the end of the file is reached for MAIL. Control is returned to the calling program, INMAIL.

Following is the code for the RPG program, REMOVE:

```

H** RPG OBJECT:          REMOVE
H** SOURCE FILE:        QRPGRSRC
H** SOURCE LIB:         OFCLIB
H**
H** DESCRIPTION: REMOVE RECORDS FROM THE FILE "OVERLMT" WHEN
H**                USERS ARE NO LONGER DELINQUENT WITH TOO MANY
H**                ITEMS IN THE INCOMING MAIL
H**
H** ABNORMAL
H** TERMINATION: RE-RUN THE CL PROGRAM DRIVING THIS PROGRAM
H**                TO CLEAR THE OUTPUT FILES AND TRY AGAIN
H**
H*****
H**                SUBROUTINE SUMMARY
H*****
H** *PSSR - STANDARD ERROR HANDLING ROUTINE
H*****
H**                INDICATOR USE
H*****
H** 10 - EOF NEW
H** 11 - NOT FOUND IN 'OVERLMT'
H*****
H** UPDATE FORMAT FOR THIS PGM YYMMDD - DEBUG=1 FOR DUMP CAPAB
H*****
H    1 Y
F*****
F* FILE WITH DELINQUENT ENTRIES
F*****
FOVERLMT UF E          K          DISK
F                                KINFSR *PSSR
F*****
F* FILE WITH DELINQUENTS TO BE DELETED BUT NO FOUND
F*****
FFGND  0 E          DISK
F                                KINFSR *PSSR
E*****
I*****
IERRDS  SDS
I                                *STATUS ESTS1
I                                *ROUTINE ROUT1
I                                *PARMS PARM1
I                                *PROGRAM PGMNAM
I                                81 90 PGMLIB
I                                244 253 JOBNAM
I                                254 263 USRNAM
I                                264 2690JOBNUM
I                                282 2870JOBTIM
I*

```

Figure 4-26 (Part 1 of 2). REMOVE RPG Program

```

C*
C*
C*
C*****
C* REMOVE ENTRIES FROM OVERLMT FILE
C*****
C*
C      *ENTRY   PLIST
C              PARM           GUSR   8
C*
C      GUSR     CHAINOVERLMTRF           11   NOT FOUND
C*
C      *IN11    IFEQ '1'
C              WRITEGND                   RCD NOT FND
C*                                     IN OVERLMT
C*
C              ELSE
C              DELETOVERLMTRF           12
C              ADD 1           TOTDLT 70   DELETED RCD
C              END
C*
C              RETRN
C*
C*
C              MOVE '1'           *INLR
C              RETRN
C*
C*
C*****
C* PSSR          * - PROGRAM ERROR SUBROUTINE
C*****
C* TO DUMP DEBUG MUST BE EQUAL TO 1 (SEE H SPEC)
C*****
C      *PSSR     BEGSR
C      PGMNAM    DUMP
C              MOVE '1'           *INLR
C              RETRN
C              ENDSR
C*****

```

Figure 4-26 (Part 2 of 2). REMOVE RPG Program

REMOVE is called by INMAIL to remove a user from the OVERLMT file. OVERLMT is declared as update, externally described, keyed, and residing in the disk. OVERLMT (CHAIN) is searched for a record with the passed user name. If found, the user name is removed. Control then returns to the calling program.

Building a Bridge between Office and Object Distribution

The entries in the system distribution directory are for office distribution and object distribution. Some users use only one kind of distribution. When you are sent a file through object distribution, but are normally an office user, you do not know the file has been sent. If a non-office user receives an office distribution, the user does not know that mail is waiting to be received.

To solve this problem, you could set an indicator in the system distribution directory that tells whether a user is an office distribution or object distribution user.

To see if an office user has files in the queue, WRKNETF is done to an output file. For the non-office user, a QRYDST is done to see if office distributions exist. If these users have files or office distribution, you can send a message, receive the files or office distribution, or delete the file if that is what you want your application to do. For our example, QRYDST sends a message to inform the user of the files or office distributions.

Assumptions

The program assumes an indicator somewhere that tells whether the user is an office user or an object distribution user. The program below also assumes a text field in the system distribution directory is set to OFFICE for the office user and OBJECT for the object user. The indicator can be done any way you want. Use this program as a starting point. You can change it to meet your requirements.

Code

The program does a DSPDIR to an output file.

If you run this program several times, performance improves if you do the following:

- Periodically do the DSPDIR to the output file QTEMP/DSPDIR DETAIL(*FULL) outside the program for updates.
- Comment out the DSPDIR and the DLTF commands in the program.

Each record in the file is read sequentially looking for the characters OFFICE or OBJECT in the text field in the directory. If OFFICE is found, the user is an office user. The program CHKNETF is called to determine whether net files are in the user's queue.

If OBJECT is found, the user is an object distribution user. The program CHKDSTF is called to determine whether office distribution files are present.

The following program, CHKDST, is the main program called:

```
PGM
/* This program checks to see if users have */
/* distributions that are either Office distributions */
/* or Object distributions. If the user is an office */
/* user and has object distribution files a message */
/* is sent to that user to do a WRKNETF command. If */
/* the user is an object distribution user and has */
/* office distribution, a message is sent to tell */
/* that user to use office. */
/* */
/* Need to have security officer authority AND user */
/* permission to all the user profiles that will be */
/* accessed (use GRTUSRPMN to get user permission). */
/* */
/* The program does a DSPDIR to an outfile. It checks */
/* the text field to see if the user is an "OFFICE" */
/* or a "OBJECT" user. This field needs to be set */
/* manually through CHGDIRE or WRKDIR commands. */
/* */
/* If the user is an "OFFICE" user, the program */
/* CHKNETF is called. This program does a WRKNET to */
/* an outfile. If there are any entries in the */
/* outfile, then a message is sent to that user that */
/* they have object distribution and to run WRKNETF. */
/* The program could be enhanced here to either */
/* receive (RCVNETF) or delete (DLTNETF) each net */
/* file in the WRKNETF outfile. */
/* */
/* If the user is an "OBJECT" user, the program */
/* CHKDSTF is called. This program does a QRYDST to */
/* an outfile. If there are any entries in the */
/* outfile, then a message is sent to that user that */
/* they have office distribution and to run office. */
/* The program could be enhanced here to either */
/* receive (RCVDST) or delete (DLTDST) each office */
/* distribution in the QRYDST outfile. */
/* */
/* The reason for having more than one program is */
/* because the called programs access different */
/* files. */
/* */
/* For example if you want to check if there are */
/* distributions for the users on your system, */
/* you would issue the following command: */
/*     CALL CHKDST */
/* */
```

Figure 4-27 (Part 1 of 3). CHKDST Program

```

/* To create this program, use the CRTCLPGM command. */
/* This is the source code for the program.          */
/*                                                    */
DCL      VAR(&SAVUSRID) TYPE(*CHAR) LEN(10) +
        VALUE('          ') /* Set saved user +
        ID value to blanks                                     */
DCL      VAR(&SAVADDR) TYPE(*CHAR) LEN(10) +
        VALUE('          ') /* Set saved address +
        ID value to blanks                                     */

DCLF     FILE(QSYS/QAOSDIRO) RCDfmt(OSDIRE) /* System +
        DSPDIR outfile layout                               */
/* Do a DSPDIR to an outfile in library QTEMP          */
DSPDIR   OUTPUT(*OUTFILE) +
        OUTFILE(QTEMP/DSPDIR) DETAIL(*FULL)

/* Now override the system file with the file just   */
/* created by DSPDIR.                                 */
OVRDBF   FILE(QAOSDIRO) TOFILE(QTEMP/DSPDIR)

/* Read the records sequentially from the DSPDIR     */
/* outfile.                                          */
READF:   RCVF      RCDfmt(OSDIRE)

/* Monitor for EOF message                           */
MONMSG   MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* If the user ID name and the address is the same as */
/* the previous entry, then skip this entry.          */
IF       COND(&WOSDDEN *EQ &SAVUSRID *AND +
             &WOSDDGN *EQ &SAVADDR) +
        THEN(GOTO CMDLBL(READF))

/* Set the saved user ID name to the current entry.  */
CHGVAR   VAR(&SAVUSRID) VALUE(&WOSDDEN)
/* Set the saved address to the current entry.        */
CHGVAR   VAR(&SAVADDR) VALUE(&WOSDDGN)

/* If the text field indicates an Office user, call  */
/* the program CHKNETF to see if the user has any    */
/* net files and if so, issue a message to the user. */
IF       COND(%SST(&WOSDTEXT 1 6) *EQ +
             'OFFICE') THEN(CALL PGM(CHKNETF) +
             PARM(&WOSDDEN &WOSDDGN &WOSDUSRP))

```

Figure 4-27 (Part 2 of 3). CHKDST Program

```

/* If the text field indicates an Object dist,          */
/* call the program CHKDSTF to see if the user         */
/* has any office distributions.                       */
ELSE +
    IF      COND(%SST(&WOSDTEXT 1 6) *EQ +
              'OBJECT') THEN(CALL PGM(CHKDSTF) +
              PARM(&WOSDDEN &WOSDDGN &WOSDUSRP))

/* Read another record from the DSPDIR outfile.        */
GOTO      CMDLBL(READF)

EOF:      /* Delete the temporary DSPDIR outfile.      */
          DLTF      FILE(QTEMP/DSPDIR)

          /* Delete the temporary WRKNETF outfile      */
          DLTF      FILE(QTEMP/WRKNETF)

          /* Monitor for file not found message      */
          MONMSG    MSGID(CPF2105)

          /* Delete the temporary QRYDST outfile      */
          DLTF      FILE(QTEMP/QRYDST)

          /* Monitor for file not found message      */
          MONMSG    MSGID(CPF2105)

ENDPGM

```

Figure 4-27 (Part 3 of 3). CHKDST Program

The program CHKDSTF is called by CHKDST:

```

PGM          PARM(&USRID &ADDR &USRP)
/* This program checks to see if the specified */
/* user has any Office distribution, and if they do */
/* sends a message to them informing them of the */
/* Office distribution they have. */
/* */
/* This program is called by CHKDST and the input is: */
/* &USRID - User ID to check for Office Dist */
/* &ADDR - Address to check for Office Dist */
/* &USRP - User profile name of the user */
/* */
/* The output is either a message to the user that */
/* they have one or more Office Distribution OR just */
/* return to CHKDST if the user does not have any */
/* Office distribution. */
/* */
/* To create this program, use the CRTCLPGM command. */
/* This is the source code for the program. */
/* */
DCL          VAR(&USRID) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADDR) TYPE(*CHAR) LEN(8)
DCL          VAR(&USRP) TYPE(*CHAR) LEN(10)
DCLF         FILE(QSYS/QAOSILIN) RCFMT(OSLIN) /* +
            System QRYDST outfile layout */

/* Do a QRYDST to an outfile in library QTEMP */
QRYDST      USRID(&USRID &ADDR) OUTFILE(QTEMP/QRYDSTIN)

/* Now override the system file with the file just */
/* created by QRYDST. */
OVRDBF     FILE(QAOSILIN) TOFILE(QTEMP/QRYDSTIN)

/* Attempt to read a record from the QRYDSTIN outfile.*/
/* If EOF message (CPF0864) is returned on the first */
/* read, then the file is empty and the user does not */
/* have any office distribution. Otherwise the user */
/* has one or more office distribution, so a message */
/* is sent to that user. */
READF:     RCVF          RCFMT(OSLIN)

/* Monitor for EOF message */
MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* There is at least one office distribution in the */
/* QRYDST outfile. Send a message to the user. */
SNDPGMMSG  MSG('You have office distribution files. +
            Issue the STROFC command with option 2 +
            to work with office distribution') +
            TOUSR(&USRP)

EOF:       /* Return to the calling program CHKDST. */
RETURN

ENDPGM

```

Figure 4-28. CHKDSTF Program Called by the CHKDST Program

The program CHKNETF is called by CHKDST:

```

PGM          PARM(&USRID &ADDR &USRP)
/* This program checks to see if the specified      */
/* user has any net files and if they do, sends     */
/* a message to them informing them of the net files.*/
/*                                                  */
/* This program is called by CHKDST and the input is:*/
/*   &USRID - User ID of office user                */
/*   &ADDR  - Address of office user                */
/*   &USRP  - User profile to check for net files   */
/*                                                  */
/* The output is either an office msg to the user that*/
/* they have one or more net files OR just return   */
/* to CHKDST if the user does not have net files.   */
/*                                                  */
/* To create this program, use the CRTCLPGM command.*/
/* This is the source code for the program.         */
/*                                                  */
DCL          VAR(&USRID) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADDR) TYPE(*CHAR) LEN(8)
DCL          VAR(&USRP) TYPE(*CHAR) LEN(10)
DCLF        FILE(QSYS/QANFDNTF) RCDFMT(QNFDNTF) /* +
           System WRKNETF outfile layout          */

/* Do a WRKNETF to an outfile in library QTEMP      */
WRKNETF     USER(&USRP) +
           OUTPUT(*OUTFILE) OUTFILE(QTEMP/WRKNETF)

/* Now override the system file with the file just  */
/* created by WRKNETF.                               */
OVRDBF     FILE(QANFDNTF) TOFILE(QTEMP/WRKNETF)

/* Attempt to read a record from the WRKNETF outfile.*/
/* If EOF message (CPF0864) is returned on the first */
/* read, then the file is empty and the user does not */
/* have any net files. Otherwise the user has one    */
/* or more net files, so a message is sent to that  */
/* user.                                             */
READF:     RCVF          RCDFMT(QNFDNTF)

/* Monitor for EOF message                          */
MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* There is at least one file in the WRKNETF outfile.*/
/* Send a distribution message to the user since this */
/* user is an office user.                          */
SNDDST     TYPE(*MSG) TOUSRID((&USRID &ADDR)) +
           DSTD('User has object distribution files') +
           MSG('You have object distribution files. +
           Issue the WRKNETF command to work with +
           those net files')

EOF:       /* Return to the calling program CHKDST.   */
RETURN

ENDPGM

```

Figure 4-29. CHKNETF Program Called by the CHKDST Program

Identifying New Mail

To find out if a user has new mail waiting, run the following program. This program can be changed, depending on your needs, to show a message on a display, to be run every so often, and also to run for all users on the system. Figure 4-30 shows the coding:

```
PGM          PARM(&USRID &ADDR &MSGQ)
/* This program checks to see if there is any new */
/* mail waiting for the specified user. If there is */
/* any mail with 'New' or 'Unopened' status in */
/* the user's incoming mail log, the message 'New */
/* mail waiting' is sent to the work station message */
/* queue specified. */
/* */
/* For example, for user CHARLIE at address CHICAGO */
/* and the message queue is CHARLIE1, you */
/* would issue the following from the command line: */
/* CALL NEWMAIL PARM(CHARLIE CHICAGO CHARLIE1) */
/* */
/* The output is either a message to the user that */
/* they have new mail or no message at all. */
/* */
/* This program is set up to be called from another */
/* program or command. It may also be run as a */
/* background job. */
/* */
/* NOTE: This program will not work with Release 1.0 */
/* or Release 1.2 because there is not a mail */
/* log item status byte in the QRYDST outfile. */
/* */
DCL          VAR(&USRID) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADDR) TYPE(*CHAR) LEN(8)
DCL          VAR(&MSGQ) TYPE(*CHAR) LEN(10)
DCL          VAR(&FILENAM) TYPE(*CHAR) LEN(10)
DCLF        FILE(QSYS/QAOSILIN) RCDfmt(OSLIN) /* +
System QRYDST outfile layout */

/* Do a QRYDST to an outfile in library QTEMP */
/* Make the outfile name unique for each user in case */
/* this program is used for multiple users. */
CHGVAR      VAR(&FILENAM) VALUE('QY' *CAT &USRID)
QRYDST      USRID(&USRID &ADDR) OUTFILE(QTEMP/&FILENAM)

/* Now override the system file with the file just */
/* created by QRYDST. */
OVRDBF      FILE(QAOSILIN) TOFILE(QTEMP/&FILENAM)
/* Read a record from the outfile. */
READF:      RCVF          RCDfmt(OSLIN)

/* Monitor for EOF message */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
```

Figure 4-30 (Part 1 of 2). NEWMAIL Program

```

/* Check the incoming mail log entry status and if */
/* the mail entry is 'New' or 'Unopened' status */
/* then issue the message and exit. */
/* In OfficeVision/400 the message */
/* 'New mail waiting' */
/* is displayed in this circumstance. If you want */
/* to only send the message when there is 'New' mail */
/* just check for &LINMSL *EQ '1'. */
/* The status of a mail item gets changed from 'New' */
/* to 'Unopened' if the user views the mail log entry */
/* in OfficeVision/400 but doesn't actually view the */
/* contents. In the case of viewing the contents, */
/* the status gets changed to 'Opened'. */
IF COND(&LINMLS *EQ '1' *OR &LINMLS *EQ '2') THEN( +
DO)
/* Sending the break message is just an example. */
/* This may be changed to send the message the way */
/* your application wants it. */
SNDBRKMSG MSG('New Mail Waiting') +
TOMSGQ(*LIBL/&MSGQ)
GOTO CMDLBL(END)
ENDDO

/* Process the next record in the outfile */
GOTO CMDLBL(READF)

EOF:
END: RETURN
ENDPGM

```

Figure 4-30 (Part 2 of 2). NEWMAIL Program

Deleting Outgoing Mail

This program deletes outgoing mail for which confirmation of delivery was requested for one or all local users. To create the files in this program, use the following commands:

```

CRTDUPOBJ OBJ(QAOSDIRO) FROMLIB(QSYS) OBJTYPE(*FILE)
TOLIB(QGPL) NEWOBJ(SYSDIR)
CRTDUPOBJ OBJ(QAOSILOT) FROMLIB(QSYS) OBJTYPE(*FILE)
TOLIB(QGPL) NEWOBJ(OUTMAIL)

```

QAOSIDIRO and QAOSILOT are model files shipped with the system. The QAOSIDIRO file is the directory output file. The QAOSILOT file is the output file created by the QRYDST command for outgoing mail. For details of the layout of these files, see Appendix C, "Office Services Output File Considerations."

Following is the command definition object used to create the Delete Outgoing Mail (DLTOUTML) CL command:

```

/*****/
/*                                          */
/*  COMMAND NAME: DLTOUTML                */
/*                                          */
/*  COMMAND TITLE: Delete Outgoing Mail   */
/*                                          */
/*  DESCRIPTION: This command deletes all  */
/*                outgoing mail for       */
/*                one or all local users  */
/*                on the system for mail   */
/*                items that are older    */
/*                than the specified      */
/*                deletion date. This    */
/*                command allows you to   */
/*                specify the user that  */
/*                you will be working on  */
/*                behalf of.              */
/*                This command was        */
/*                created with the        */
/*                intention that the      */
/*                system administrator or */
/*                security officer will   */
/*                start this for other    */
/*                users.                  */
/*                                          */
/*                                          */
/*****/
DLTOUTML: CMD  PROMPT('Delete Outgoing Mail')
          PARM  KWD(USRID) +
            TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) +
            MAX(1) FILE(*NO) FULL(*NO) PASSATR(*NO) +
            SPCVAL(*ALL) +
            EXPR(*YES) VARY(*NO) +
            PROMPT('Remove mail for - userid')
          PARM  KWD(ADDR) +
            TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(0) +
            MAX(1) FILE(*NO) FULL(*NO) PASSATR(*NO) +
            EXPR(*YES) VARY(*NO) +
            PROMPT('Remove mail for - address')
          PARM  KWD(DATE) +
            TYPE(*CHAR) LEN(6) RSTD(*NO) MIN(1) +
            MAX(1) FILE(*NO) FULL(*NO) PASSATR(*NO) +
            EXPR(*YES) VARY(*NO) +
            PROMPT('Remove Mail Date - MMDDYY')
          PARM  KWD(GRNTTO) +
            TYPE(*CHAR) LEN(10) RSTD(*NO) MIN(1) +
            MAX(1) FILE(*NO) FULL(*NO) PASSATR(*NO) +
            EXPR(*YES) VARY(*NO) +
            PROMPT('Grant permission to-USRPRF')

```

Figure 4-31. DLTOUTML Command

Following is the source for the CRTDIRFILE CL program:

```

/* ***** */
/* Program Name : CRTDIRFILE */
/* Program Description: This program creates the file that */
/*                      contains the system distribution */
/*                      directory for one or all local users. */
/*                      This program calls the FNDUSER CL */
/*                      Program. */
/* ***** */
PGM      PARM(&USR &ADD &DATE &GRNTTO) /* Create +
        directory outfile records */
DCL      VAR(&USR) TYPE(*CHAR) LEN(8) /* Remove +
        mail user ID */
DCL      VAR(&ADD) TYPE(*CHAR) LEN(8) /* Remove +
        mail user address */
DCL      VAR(&DATE) TYPE(*CHAR) LEN(6) /* +
        Remove mail date */
DCL      VAR(&GRNTTO) TYPE(*CHAR) LEN(10) /* Grant +
        user permission to - user profile */
DCL      VAR(&ALL) TYPE(*CHAR) LEN(8) VALUE('*ALL ')
/* Display the directory information to an outfile (DSPDIR) */
IF      COND(&USR *EQ &ALL) THEN( +
DO)
        DSPDIR      USRID(*ALL) OUTPUT(*OUTFILE) +
        OUTFILE(QGPL/SYSDIR) DETAIL(*FULL)
        MONMSG      MSGID(CPF0000) EXEC(GOTO CMDLBL(OUT))
ENDDO
ELSE +
DO
        DSPDIR USRID(&USR &ADD) OUTPUT(*OUTFILE) +
        OUTFILE(QGPL/SYSDIR) DETAIL(*FULL)
        MONMSG      MSGID(CPF0000) EXEC(GOTO CMDLBL(OUT))
ENDDO
CALL    PGM(*LIBL/FNDUSER) PARM(&USR &ADD &DATE +
        &GRNTTO) /* Find the user from the +
        directory output file to do a query +
        distribution for */
MONMSG      MSGID(CPF0000) EXEC(GOTO CMDLBL(OUT))

OUT:
ENDPGM /* End of Program Create Directory File (CRTDIRFILE) */

```

Figure 4-32. CRTDIRFILE Program

Following is the source for the FNDUSER CL program:

```

/* ***** */
/* Program Name: FNDUSER */
/* Program Description: This program loops through the file */
/*                      SYSDIR in QGPL that was created */
/*                      by the CRTDIRFILE Program. All mail is */
/*                      processed for all local users. */
/*                      The *ANY entries are bypassed. All */
/*                      other entries are processed. */
/*                      All remote entries are determined */
/*                      when the GRTUSRPMN command is run. */
/*                      This program calls the CRTOUTFL CL */
/*                      Program. */
/* ***** */
PGM          PARM(&USR &ADD &DATE &GRNTTO)
/* Declare the files and variables used by the program */
DCLF         FILE(QGPL/SYSDIR) /* FILE SYSDIR +
              contains the records resulting from the +
              DSPDIR. It has a record format name OSDIRE.*/
DCL         VAR(&ANYID) TYPE(*CHAR) LEN(8) VALUE('*ANY +
              ') /* *Any user ID */
DCL         VAR(&USR) TYPE(*CHAR) LEN(8) /* Remove +
              mail for user ID */
DCL         VAR(&ADD) TYPE(*CHAR) LEN(8) /* Remove +
              mail for user address */
DCL         VAR(&DATE) TYPE(*CHAR) LEN(6) /* +
              Remove mail date */
DCL         VAR(&GRNTTO) TYPE(*CHAR) LEN(10) /* +
              Grant user permission to user profile */
DCL         VAR(&GRNTFOR) TYPE(*CHAR) LEN(10) /* +
              Grant user permission for user profile */
RCV:        RCVF         RCDfmt(OSDIRE)
              MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(OUT))
              IF COND(&WOSDDEN *NE &ANYID) THEN( +
              DO)
              CHGVAR     VAR(&USR) VALUE(&WOSDDEN)
              CHGVAR     VAR(&ADD) VALUE(&WOSDDGN)
              CHGVAR     VAR(&GRNTFOR) VALUE(&WOSDUSRP)
              CALL       PGM(CRTOUTFL) PARM(&USR &ADD &DATE &GRNTTO +
              &GRNTFOR) /* Query user outgoing mail */
              ENDDO
              GOTO       CMDLBL(RCV)

OUT: RETURN
ENDPGM /* End of Find User Program (FNDUSER) */

```

Figure 4-33. FNDUSER Program

Following is the source for the CRTOUTFL program:

```

/* ***** */
/* Program Name: CRTOUTFL */
/* Program Description: This program queries a specific user's outgoing mail. This information is written to an output file. Permission is granted for the specified user to work on behalf of another user. At the end of the program, the user's permission is revoked. This program calls the DLTMAIL CL Program.
/* ***** */
PGM      PARM(&USR &ADD &DATE &GRNTTO &GRNTFOR)
DCL      VAR(&USR) TYPE(*CHAR) LEN(8)
DCL      VAR(&ADD) TYPE(*CHAR) LEN(8)
DCL      VAR(&DATE) TYPE(*CHAR) LEN(6)
DCL      VAR(&GRNTTO) TYPE(*CHAR) LEN(10)
DCL      VAR(&GRNTFOR) TYPE(*CHAR) LEN(10)
/* Grant user permission to &GRNTTO for &GRNTFOR */
GRTUSRPMN  TOUSER(&GRNTTO) FORUSER(&GRNTFOR)
MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(GRNTFAIL))
/* Query user outgoing mail and delete outgoing status */
QRYDST     OPTION(*OUT) USRID(&USR &ADD) DLTSTS(*YES)
MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(OUT))
/* Query user outgoing mail, put data in outfile */
QRYDST     OPTION(*OUT) USRID(&USR &ADD) +
           OUTFILE(QGPL/OUTMAIL)
MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(OUT))
CALL       PGM(*LIBL/DLTMAIL) PARM(&USR &ADD &DATE)
MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(OUT))

OUT:
/* Revoke user permission */
RVKUSRPMN  FROMUSER(&GRNTTO) FORUSER(&GRNTFOR)
MONMSG     MSGID(CPF0000) EXEC(GOTO CMDLBL(GRNTFAIL))
GRNTFAIL:
ENDPGM /* End of Create Outgoing Mail Information Program */

```

Figure 4-34. CRTOUTFL Program

Following is the source for the DLTMAIL program:

```

/* ***** */
/* Program Name: DLTMAIL */
/* Program Description: This program deletes all outgoing */
/* mail for a specific user. */
/* ***** */
/* This program deletes mail items from the outgoing mail for the */
/* user "&USR &ADD &DATE" */
      PGM          PARM(&USR &ADD &DATE)
/* Declare the files and variables used by the program */
      DCLF        FILE(QGPL/OUTMAIL) /* File MAIL +
      contains the records resulting from the +
      QRYDST. It has a record format name OSLOUT.*/
      DCL         VAR(&USR) TYPE(*CHAR) LEN(8)
      DCL         VAR(&ADD) TYPE(*CHAR) LEN(8)
      DCL         VAR(&DATE) TYPE(*CHAR) LEN(6)
      DCL         VAR(&COUNT) TYPE(*DEC) LEN(6 0)
      DCL         VAR(&PREVDID) TYPE(*CHAR) LEN(20)
      DCL         VAR(&MM) TYPE(*CHAR) LEN(2)
      DCL         VAR(&DD) TYPE(*CHAR) LEN(2)
      DCL         VAR(&YY) TYPE(*CHAR) LEN(2)
      DCL         VAR(&DLTDATE) TYPE(*CHAR) LEN(6)
      DCL         VAR(&MAILDATE) TYPE(*CHAR) LEN(6)

/* Delete all the status for the user */
/* Then delete all outgoing(COD) entries */
/*
      CHGVAR      VAR(&COUNT) VALUE(0)
RCV:           RCVF      RCDFMT(OSLOUT)
              MONMSG    MSGID(CPF0864) EXEC(GOTO CMDLBL(OUT))

      IF COND(&COUNT *GT 0) THEN( +
      DO)
        IF COND(&PREVDID *EQ &OUTDID) THEN(+
        DO)
          CHGVAR      VAR(&COUNT) VALUE(&COUNT + 1)
          GOTO CMDLBL(RCV)
        ENDDO
      ENDDO
      CHGVAR      VAR(&MM) VALUE(%SST(&DATE 1 2))
      CHGVAR      VAR(&DD) VALUE(%SST(&DATE 3 2))
      CHGVAR      VAR(&YY) VALUE(%SST(&DATE 5 2))
      CHGVAR      VAR(&DLTDATE) VALUE(&YY *CAT *MM *CAT &DD)
      CHGVAR      VAR(&MAILDATE) VALUE(%SST(&OUTSDT 3 6))
      IF COND(&MAILDATE *LE &DLTDATE) THEN(+
      DO)
        DLTST      DSTID(&OUTDID) OPTION(*OUT) USRID(&USR &ADD)
        MONMSG      MSGID(CPF0000)
      ENDDO
      CHGVAR      VAR(&PREVDID) VALUE(&OUTDID)
      CHGVAR      VAR(&COUNT) VALUE(&COUNT + 1)
      GOTO        CMDLBL(RCV)

OUT: RETURN
ENDPGM /* End of Deleting Outgoing Mail */

```

Figure 4-35. DLTMAIL Program

Chapter 5. Document Library Services

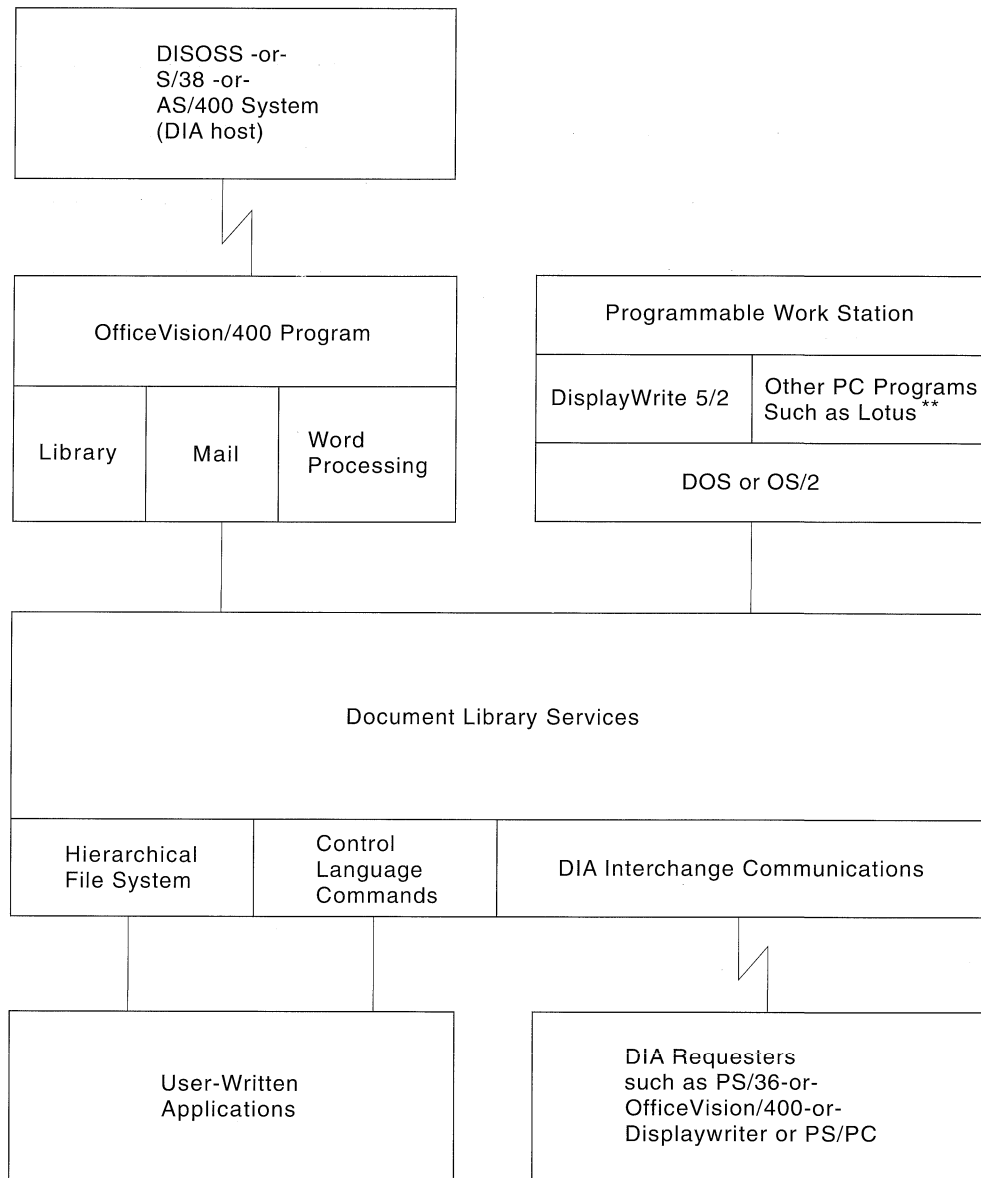
This chapter describes document library services and how they relate to documents and folders. This chapter also lists control language (CL) interfaces and provides programming examples.

Document Library Services Overview

Document library services is the set of functions used to manage documents and their related objects. These functions include:

- Creating and filing documents into the library
- Creating and managing folders
- Storing and managing document and folder details
- Searching for documents based on document details
- Organizing documents into folders
- Organizing documents into lists based on user-specified search values
- Copying documents
- Authorizing users to documents and folders
- Saving and restoring documents and folders
- Deleting documents and folders

Figure 5-1 shows the organization of the interfaces between AS/400 functions and document library services.



RV2W213-0

Figure 5-1. AS/400 System Access to the Document Library

Accessing Document Library Services

AS/400 documents are stored in a common storage place and are shared among users of the **document library**. Document library users include:

- Host display station users, such as users of 3180 devices
- Local PC users
- Remote users

Sharing documents is enhanced by functions that transform one document content data stream to another when a document is retrieved or accessed.

AS/400 functions that use document library services are summarized in Figure 5-2.

Figure 5-2. AS/400 Functions Using Document Library Services

| Function | Use |
|----------------------------------|---|
| OfficeVision/400 word processing | Used for document-related word processing functions such as edit, view, and print. Also used to store nontext-related information such as charts and images. |
| OfficeVision/400 library/mail | Used for: <ul style="list-style-type: none"> • Filing documents into the DIA library • Storing documents retrieved by remote document library services • Searching and retrieving documents • Changing document details |
| Control language | Used to program additional customer-defined library access functions. Functions are supported to allow documents to be filed, retrieved, deleted, searched, renamed, moved, or changed. Also used to administer the document library. |
| DisplayWrite* program | Used for document-related word processing functions such as edit, view, and print on the personal computer. |
| Hierarchical File System (HFS) | Used by application programs to interface with documents and folders. The HFS allows access to the document library services file system functions and other file systems with a common interface. See the <i>System Programmer's Interface Reference</i> manual and Appendix E, "Hierarchical File System (HFS) and DLS" on page E-1 for additional information. |
| Other PC programs | Store office and non-office related files in folders. |
| Remote DIA requesters | File, search, retrieve, print, and change documents in the host library. |

Host Display Station Interface

Host display station users may access all the documents, folders, and document lists stored in the library. OfficeVision/400 includes library and mail handling functions that access these objects in the library. The OfficeVision/400 word processing function stores text and nontext data in the library. OS/400 functions include managing as well as filing, retrieving, searching, and changing the objects in the document library. These functions can be accessed through the OfficeVision/400 interface or OS/400 CL commands.

To start OfficeVision/400, or to access the OfficeVision/400 word processing function, use one of the following commands:

STROFC The Start Office command displays the OfficeVision/400 menu.

Note: An application can call specific functions of STROFC directly. For example, typing STROFC 5 bypasses the OfficeVision/400 main menu and goes directly to the Work with Documents and Folders display.

STRWP The Start Word Processing command starts the OfficeVision/400 text editor and displays the Word Processing menu.

The following command menus are available for office services:

- CMDOFC** The Office Commands menu lists office-related menus.
- CMDDOC** The Document Commands menu lists document-related commands.
- CMDFLR** The Folder Commands menu lists folder-related commands.
- CMDDLO** The Document Library Object Commands menu lists document-library-object-related commands.

To access the command menus, type `go` and the menu name. For example, to see a menu of document library object related commands, type:

```
go cmddlo
```

Remotely Attached System Interface

Remotely attached applications access the library using the functions defined by the Document Interchange Architecture (DIA). An AS/400 system provides DIA host document library services support to remote system applications such as Personal Services/36, Displaywriter, Personal Services/PC, or another AS/400 system. As a result, all documents created on the host system or a personal computer can be filed, searched, retrieved, changed (if the remote application is on another AS/400 system), or deleted. Folders cannot be accessed by a remotely attached application. Also, the following object-management functions are available when accessed by another AS/400 system:

- Certain document authorization functions. These functions are listed under “Remotely Filed Documents” on page 6-15.
- Checking out a document allows a remotely attached system to lock a document and prevent any additional updates until the document is checked back in. Documents can be checked out indefinitely, including across system work stations.

Locally Attached PC Interface

Locally attached PC applications view the document library as a set of PC files and subdirectories. The locally attached PC applications access the subdirectories using PC Support/400. PC Support/400 redirects DOS file I/O and OS/2 file I/O requests to the host for processing. Documents not stored in folders are not available when you access the library using PC Support/400. Documents can be checked out and checked in through PC Support/400. Document details can be accessed through OS/2 extended attributes and PC Support/400.

Document Library Services Objects

The document library can contain documents and folders. The documents and folders are called **document library objects**.

Figure 5-3 shows the relationship of the document library objects.

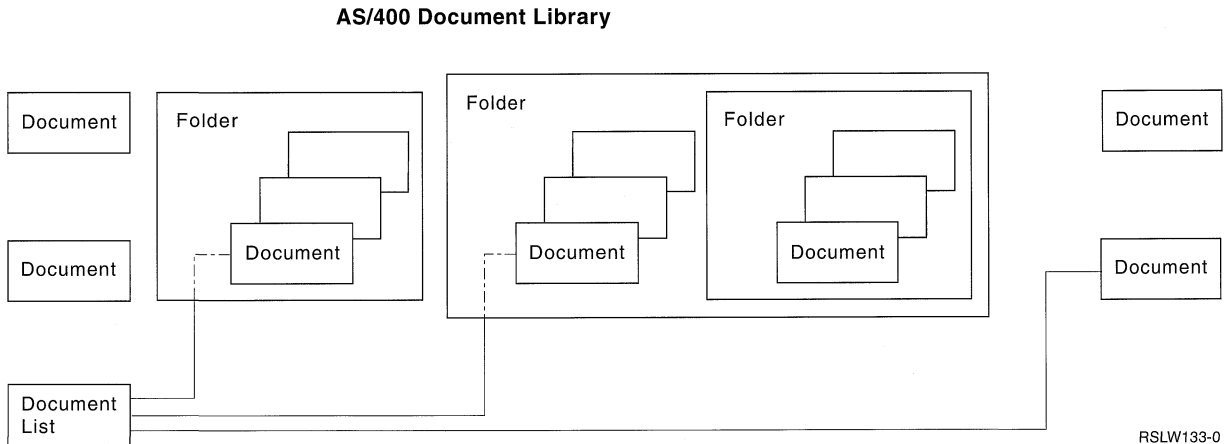


Figure 5-3. AS/400 Document Library Objects

Documents

Each document in an AS/400 system is stored in a separate system object. A document has two logical parts:

Document content

The text portion of the document. You can change the document content at any time if replacement is allowed. You can edit a document created by the OfficeVision/400 word processing function, for example, unless it was changed to not replaceable. You may designate a document replaceable or not replaceable when you file the document.

Document details

Items such as description, subject, author, and other fields that you can use to search for or identify the document. Document details are also referred to as *document attributes*.

You can change document details at any time for documents that are either replaceable or not replaceable. You can specify the details of a document when it is created. To change document details, use the Change Document Description (CHGDOCD) or Start Office (STROFC) commands or Option 8 (Details) on displays that can work with documents.

You can do the following to documents:

- Copy
- Delete
- Rename
- Reorganize storage
- Reclaim
- Replace
- Retrieve
- Print
- Change details
- Change authority
- Check for existence
- File a document locally
- Search for documents
- Check documents in and out
- Restore and save

With OfficeVision/400, you can also do the following to documents:

- Revise
- View
- Spell check
- Paginate
- Create documents with the editor
- File documents to a remote location

See Chapter 8, “Word Processing Services,” for more information about other commands that work on documents and library objects.

Folders and Document Lists

Documents within the library can be organized by using either document lists, folders, or both. Each document list and folder is a separate system object.

Users may group documents from the library into document lists by using the search function to create document lists. Input to the search are user-defined search values, such as subject and date written. A **document list** is static, which means after the document list is created, document references cannot be added to or removed from it. The search values are stored in the document list so the search can be run again to get a more current list of items. A document list created with the QRYDOCLIB command does not store the search values when using the Change criteria, Create criteria, View criteria, or Run search. A document list is itself a document in the document library. Therefore, the rules that apply to documents also apply to document lists. Most of the functions that are supported for documents can also be used for document lists. Each document list is a separate system object.

The following functions are supported for document lists:

- Create a document list
- Delete one or more document lists
- Create an output file which contains the results of a search
- Copy or move lists into folders
- Save documents in a document list

The following functions are supported for document lists by OfficeVision/400:

- Create a document list
- Display all document lists owned by a user
- Display all document lists authorized to the user
- Display the documents within a list
- View and edit a document in the list
- Delete a document in the list

A **folder** is a directory for documents. A folder is used to group related documents and to find documents by name. Folders are user-defined and can include documents and other folders. Authorized users can add or remove individual documents and folders at any time. Documents and folders cannot be included in more than one folder, but do not need to be included in any folder. The following functions are supported. These functions can be accessed through OfficeVision/400 or through CL commands. Some of these functions may also be available through other interfaces.

- Display all folders that can be accessed by a user
- Display the documents and folders of a folder
- View, edit, or delete documents in a folder
- Copy or move a document to or from a folder
- Save or restore a folder and the documents and folders within it
- Delete a folder and the documents and folders within it
- Create an empty folder
- Rename a folder
- Reclaim a folder
- Dump a folder
- Reorganize a folder

The first-level folder is a folder name that is not preceded by another folder name. A first-level folder is the first folder name in a folder path. A first-level folder can contain both folders and documents.

The root folder is the folder on the system that contains all other folders. The system recognized identifier for a root folder is *ROOT. The root folder contains all first-level folders. It cannot contain documents.

PC Files and Folders

You can use folders to hold PC objects as well as documents and folders. The PC object is a document when stored in a folder.

Note: All folders are automatically able to be shared with PC users. No special definition or processing is needed for PC access.

The folder looks like another PC drive or subdirectory to the PC user. The PC user can store PC files in the folder. The AS/400 and PC users can work with the same folder at the same time. The AS/400 system controls access to each object in the folder. See the *PC Support/400 User's Guide for DOS* or the *PC Support/400 User's Guide for OS/2* for more information on folders.

Object Ownership

All AS/400 document library objects are owned by the users who create them.

Note: If a user is working on behalf of another user and creates a document, it is owned by the user on whose behalf the work is being done.

Document library **object ownership** can be transferred from one user to another using the CHGDLOOWN command. Refer to “Ownership Rights” on page 6-4 for more information on ownership rights, and to “Change Document Library Object Owner (CHGDLOOWN) Command” on page 6-21 for more information on transferring ownership from one user to another.

Document Library Object Naming

Documents and folders are recognized by the following:

- A user-assigned document library object name
- An OS/400 object name

User-assigned document or folder names consist of 1 to 8 characters, optionally followed by a 3-character extension. The name is unique in the containing folder or unique in the system if it is a first-level folder. To fully identify a folder or document within a folder, a path name is used. The path name is a compound name in this form:

first-level-folder/next-folder/next-folder

The system allows any single-byte EBCDIC graphic characters to appear in a folder or document name, except for the following 3 characters:

Asterisk (*)
Slash (/)
Question mark (?)

In addition, lowercase English alphabet characters (26 characters, a through z), are treated as though they were uppercase.

Note: The system does not recognize graphic characters, but only code points, and makes the following assumptions:

- All folder and document names are written in single-byte EBCDIC code pages. Consequently, only code points hex 41 through FE are allowed, since they are the EBCDIC code points used to represent graphic characters.
- Code points hex 5C, 61, and 6F are used to represent the asterisk, slash, and question mark characters as follows:

5C Asterisk (*)
61 Slash (/)
6F Question Mark (?)

Consequently, these code points are not allowed.

- Code points are converted as follows:
 - Code points hex 81 through 89 convert to code points C1 through C9.
 - Code points hex 91 through 99 convert to D1 through D9.
 - Code points hex A2 through A9 convert to E2 through E9.

These code points are used to represent the lowercase and uppercase English alphabetic characters.

For more information on code pages supported on the AS/400 system, see the *Device Configuration Guide* or the CRTDEV DSP command in the *CL Reference* manual.

All folders and documents in folders have only one user-assigned name. Documents that are not included in any folder do not have a user-assigned name. The user-assigned name is used by both the OfficeVision/400 and PC users. To access a document from the personal computer, it must have a user-assigned name and, therefore, must be in a folder.

Each document is assigned a unique identifier by the system called the **library-assigned document name (LADN)** as specified by DIA. The LADN includes a time stamp and a system name that are assigned to a document when it is filed in the document library. A 10-character OS/400 object name is created by document library services and is derived from the LADN. Syntax rules for this name are the same rules as apply to other system objects, for example, database files and programs, except that in this case the name is assigned by the system and not by the user.

Each document and folder in the library has only one system-object name. Office and PC-related functions that use the document library do not generally use this name. Non-OfficeVision/400 functions, such as the Display Object Description (DSPOBJD) CL command, can use the system object name. You can use the Display Document Library Object Name (DSPDLONAM) command to display the different identifiers for a document or folder (see “Display Document Library Object Name (DSPDLONAM) Command” on page 5-26). Use the Retrieve Document Library Object Name (RTVDLONAM) command in a program to retrieve the different identifiers (see “Retrieve Document Library Object Name (RTVDLONAM) Command” on page 5-40).

Other Document Library Services Topics

The following sections describe other document library services topics.

Enrollment Requirements

Generally, when using document library services, you must be enrolled in the system distribution directory. See descriptions of individual CL commands in this chapter for exceptions.

Document Library Security Considerations

In addition to the security provided for all objects on the AS/400 system, additional security characteristics are provided for document library objects. These characteristics are:

- Working on behalf of other users and marking objects as personal items
- Using access codes

Also, a security restriction applies to document library objects. When document library objects are checked for authority, the programs that adopt the owner's

authority are not recognized. See Chapter 6, "Security Services," for more information on these security characteristics.

Document List Considerations

Document lists are just another type of document. You can authorize other users to document lists the same way you authorize other users to documents. This means other users can share a document list in the same way they can share any document. Because document lists of remote libraries can contain attribute information, such as security, you should be careful not to authorize users to document lists with attributes you do not want them to see.

Access Codes

Access codes are supported for compatibility with DISOSS and System/38. It is recommended that you use authorization lists or group profiles instead of access codes whenever possible, because authorization lists allow users in the list to have their own level of authorization to objects associated with the list. (Access codes only permit *USE authority.)

Considerations When Working on Behalf of Another User

Other users can access the same information that you can access if you give them permission to do so. This is called working on behalf of another user. You can authorize a person to work on all your documents and folders as well as your mail by granting permission to work on your behalf, using the command:

```
GRTUSRPMN TOUSR(user-profile)
```

The person can now work on all your documents, folders, and mail.

Mark as personal those documents and folders you do not want users who are working on your behalf to access. Any object marked as personal is still available to users who are normally authorized to it, but is unavailable to users who are working on your behalf (even though it may be available to them when they are not working on your behalf).

Folder Security on the Personal Computer

A user's authority to a folder that is accessed through PC Support/400 is the same as when the folder is accessed through the AS/400 system. When a user starts PC Support/400, the user must provide a user profile that is associated with the work being done through PC Support/400. When the user requests that a folder be attached to the personal computer, the same authority checking is done as when that user is signed on to the AS/400 system. If the user does not have authority to use the folder, a message is displayed indicating that access was denied to the folder.

If the user has authority to look at the folder but not change it, the user can attach the folder and display its directory but cannot add any objects to the folder. If the user has authority to change the folder, the user can add items to the folder, but not necessarily delete or change items in the folder; the user must have the proper authority to the objects in the folder to work with the objects (such as deleting or changing items in the folder).

Character Set and Code Page Considerations

All character information for document details stored in the database search index is converted to code page 500, character set 697, whenever possible. This allows more information to be found on searches. Searches to find data that cannot be converted must specify the code page that the data is in for a match to be found. Data is not converted when it is entered from a work station that is using a character set and code page that cannot be converted to code page 500, character set 697. It is kept in its original form and tagged with the code page and character set that it was entered in. Data from remote systems must be tagged with the proper code page and character set if it is not in code page 500, character set 697.

Note: Folder and document names have no character set or code page associated with them. For validation and for setting the names all in uppercase, they are assumed to be character set 697, code page 500.

Work in Process Flag or Allow Replacement Value

All documents have an indicator that shows if there is work in process for the document content or if the document content should no longer be changed. This indicator (or flag) can be set to **revisable** or **final**. Revisable means that work is still in process on the document content and that it is available for revisions. Final means that the document content is no longer available for revisions through the editor, but other information for the document, such as document details or authorizations, can be changed.

Once a document is marked final, it cannot be changed back to revisable. You can always make copies of the document regardless of the setting of the work in process flag.

Work in process is not the same as **final-form text** (the document type of FFTDCA and FFTAS400). Final-form text indicates what form the content of the document is in, not if the document is available for editing. You can mark a document as final to prevent further accidental editing of the document. (To prevent unauthorized editing, use authorizations.) Use the Change Document Description (CHGDOCD) command to set this indicator.

Checking Out and Checking In Documents

Checking out a document means that only one user (the person who checked out the document) is allowed to make document content or document detail changes. Other changes, such as authority changes, and operations, such as saving the document, can still be done while the document is checked out.

You can check out a document locally by using the Retrieve Document (RTVDOC) command (which must be followed by the Replace Document (RPLDOC) command to check the document back in) or through PC Support/400. A document is also checked out when you get the document from a remote AS/400 system. The document is automatically checked back in when it is filed back on the remote system.

To check a document out or back in, you need *CHANGE authority to the document or all object (*ALLOBJ) or security administrator (*SECADM) special authority.

If a document whose content is not being updated is checked out, it can be released by the following people using the Edit Document Library Object Authority (EDTDLOAUT) or CHGDOCD command:

- The document owner
- A user with *ALL authority to the document
- The person who checked out the document
- A user with *ALLOBJ or *SECADM authority

Only the person who checked out a document can check it back in. The user ID and address of the person checking in the document must match the user ID and address of the person who checked out the document.

Document Library Services Control Language Interfaces

The document library services control language (CL) commands allow you to work with the AS/400 document library and the objects in it. The commands allow you to:

- Display folders (DSPFLR)
- Change details for a document (CHGDOCD)
- Work with document lists (QRYDOCLIB, DLTDOCL)
- Save, restore, reorganize, or reclaim all or part of the document library (SAVDLO, RSTDLO, RGZDLO, RCLDLO)
- Copy information into and out of objects in the library (CPYFRMPCD, CPYTOPCD, RTVDOC, FILDOC, RPLDOC)
- Check out and check in documents (RTVDOC, RPLDOC, EDTDLOAUT)
- Move documents in and out of folders and from one folder to another (MOVDOC)
- Create, delete, display, copy, rename, and retrieve object name information in the library (CRTFLR, CRTDOC, DLTDLO, DSPDLONAM, CPYDOC, RNMDLO, RTVDLONAM)

The following OS/400 CL commands are used to manage document library objects within a document or folder:

Change Document Description (CHGDOCD)

Changes the document details of a document.

Check Document Library Object (CHKDLO)

Checks for the existence of a document or folder. Optionally, checks user's authority to a document or folder.

Copy Document (CPYDOC)

Copies a document from one folder to another folder. Copies a document not in a folder to a folder. Makes another copy of a document in the same folder.

Copy From PC Document (CPYFRMPCD)

Copies the data in a PC document to a physical database file member. Optionally translates the data, for example, from ASCII to EBCDIC.

Copy To PC Document (CPYTOPCD)

Copies a member of a database file to a PC document.

Create Folder (CRTFLR)

Creates a first-level folder or a folder within a folder.

Delete Document Library Object (DLTDLO)

Deletes one or more document library objects. This command is used to reclaim space when system storage becomes critical. It is also used to remove old documents and folders that are no longer needed from the document library.

Display Document Library Object Name (DSPDLONAM)

Accepts one form of name for a document or folder object and displays other forms of the name for that object.

Display Folder (DSPFLR)

Displays information, prints information, or creates a database file with information about the documents within a folder or about the folders within a folder.

Dump Document Library Object (DMPDLO)

Produces an internal representation of folders, documents, or internal document library system objects to a spooled file. This command is used primarily for problem analysis.

File Document (FILDOC)

Creates a new document using information from a database file. The file can be user-created, or created by the RTVDOC or RCV DST command.

Move Document (MOVDOC)

Moves documents from one folder into another. Moves documents that are not in a folder into a folder. Moves documents that are in a folder out of a folder.

Query Document Library (QRYDOCLIB)

Searches for documents based on values specified by the user. References to the documents found are stored in a document list. Information about the documents can be stored in a database file.

Reclaim Document Library Object (RCLDLO)

Reclaims damaged document, folder, or internal document library system objects.

Rename Document Library Object (RNMDLO)

Changes the user-assigned name of a document library object.

Reorganize Document Library Object (RGZDLO) command

Reorganizes a document library object to improve its storage usage.

Replace Document (RPLDOC)

Replaces the content, document type, or interchange document profile (IDP) of a document. The replacement data can come from a database file created by the RTVDOC or RCV DST command or from a database file that is user defined. The document must be checked out before it can be replaced using the RPLDOC command.

Retrieve Document Library Object Name (RTVDLONAM)

Accepts one form of name for a document or folder object and retrieves other forms of the name for that object.

Retrieve Document (RTVDOC)

Copies information from a document to a database file. Optionally, checks out a document.

Restore Document Library Object (RSTDLO)

Restores one or more document library objects from diskette, tape, or a save file.

Save Document Library Object (SAVDLO)

Saves one or more document library objects to diskette, tape, or a save file.

Unless indicated, you must be enrolled in the system distribution directory to use these commands.

You must have the appropriate authority to the command to use it. See the *CL Reference* manual for a list of authorities needed to use a command, for information about the parameters for these commands, and for possible error messages created from the CL commands.

Change Document Description (CHGDOCD) Command

The Change Document Description (CHGDOCD) command changes the information describing a document that has already been filed in the document library.

Security Considerations

If USRID (*CURRENT) is specified, then the user of this command must have at least *CHANGE authority to the document whose description information is to be changed, or have *ALLOBJ special authority. If USRID (user ID address) is specified, the user of this command must have permission to work on the behalf of the user specified, that user must have at least *CHANGE authority to the document or have *ALLOBJ special authority, and the document must not be marked personal.

Some Ways to Use This Command

After a document is filed, the document details of that document can be changed with this command. For example, if you want to change a document to have a document class of MEMO, use the DOCCLS parameter.

To change a multiple number of documents, do a QRYDOCLIB. From the output file you can process the records and use the document ID to change the document description in a CL program.

Rules

Some rules that apply to this command follow:

- The document or document ID, whichever is specified, needs to exist.
- See the *CL Reference* manual for rules on each parameter for this command.

Hints and Techniques

To change the document description of a folderless document, specify the document ID. To find the document ID of a document, use the QRYDOCLIB command and specify *DOCD or *ALL for the Type of Data For Output parameter.

Character Set and Code Page Considerations

The USRID parameter is translated from the value specified in the CMDCHRID parameter to character set 930 code page 500.

The following parameters are stored in the document without translation and are considered to be in the character set and code page specified in the CMDCHRID parameter. They are also stored in the document and folder search index. If possible, they are translated to character set 697, code page 500 before being stored in the index. Otherwise, they are stored in the database without being translated and are considered to be in the character set and code page specified in the CMDCHRID parameter.

- DOCD
- AUTHOR
- DOCCLS
- KWD
- SUBJECT
- FILCAB
- CPYLIST
- REFERENCE
- STATUS
- PROJECT

Check Document Library Object (CHKDLO) Command

The Check Document Library Object (CHKDLO) command can be used to verify the existence of a document or folder. It can optionally be used to check the authority of a user to a document or folder.

Note: This command is similar to the Check Object (CHKOBJ) command, but provides added functions for document library objects. The CHKDLO command allows user-defined document and folder names. This command also checks access code authorization and checks if the document or folder is marked personal. The CHKDLO command also allows checking a user's authority when working on someone's behalf.

If the command user does not have *ALLOBJ special authority, then the command user can optionally check another user's authority to the object if the command user is authorized to work on the other user's behalf (and both are in the system directory).

If the command user has *ALLOBJ special authority, that user can check any other user's authorities to the object, even if the command user is not authorized to work on the other user's behalf. (The other-user must be in the system directory, however.) For an example of how this capability can be used, see Figure 6-3 on page 6-28.

If the object being verified is not found or the user does not have at least the authority specified on the command, an escape message is sent. When used in a CL program, one or more MONMSG commands should follow the command to monitor for any messages from the command. The following escape messages can be sent by this command:

CPF8A11 CHKDLO command failed

This message is sent for these conditions:

- The object is damaged.
- Internal objects are damaged.
- Required resources are not available.
- The user of the command does not have *ALLOBJ special authority and is not in the system distribution directory.
- The user specified in the USRID parameter is not in the system distribution directory.
- The user of the command does not have *ALLOBJ special authority and does not have permission to work on behalf of the user specified in the USRID parameter.

CPF8A75 Not authorized to folder

CPF8A75 is returned if the object is a folder and the user does not have at least the authority specified by the AUT parameter.

CPF8A77 Folder not found

CPF8A77 is returned if OBJTYPE(*FLR) is specified and the folder does not exist.

You should check for CPF8A82 if you are checking for the existence of a document library object and you do not specify the type of object by using the OBJTYPE parameter.

Note: This message is returned if the AUT parameter is (*EXCLUDE) and the user has some authority to the folder.

CPF8A82 Document not found

CPF8A82 is returned if OBJTYPE(*DOC) was specified and the document was not found or the OBJTYPE parameter was not specified and neither a document or folder by that name could be found.

CPF8A83 Not authorized to document

CPF8A83 is returned if the object is a document and the user does not have at least the authority specified by the AUT parameter.

Note: This message is returned if the AUT parameter is (*EXCLUDE) and the user has some authority to the document.

Security Considerations

You should be aware of the following security considerations:

- A user profile that has *ALLOBJ special authority or a user profile associated with a process that adopts *ALLOBJ special authority does not have to be enrolled in the system distribution directory to check another user's authority to a document library object.
- A user with no special authority can only check his own authority or the authority that he would have if working on behalf of another user. The user must have permission to work on behalf of another user before the check can be made.

Some Ways to Use This Command

Some ways to use the CHKDLO command are the following:

- Check for the existence of a first-level folder.
CHKDLO DLO(folder-name) FLR(*NONE)
- Check for the existence of a folder in a folder.
CHKDLO DLO(folder-name) FLR(folder-path) OBJTYPE(*FLR)
- Check for the existence of a document in a folder.
CHKDLO DLO(document-name) FLR(folder-path) OBJTYPE(*DOC)
- Check for the existence of a document that is not in a folder.
CHKDLO DLO(*SYSOBJNAM) SYSOBJNAM(system-name)
OBJTYPE(*DOC)
- Check if the current user has at least a specified authority to a document.
CHKDLO DLO(document-name) FLR(folder-path) AUT(*CHANGE)

Note: If the user has *CHANGE or *ALL authority or the user has *ALLOBJ special authority, then the command will complete without an exception being returned. If the user has *EXCLUDE authority or *USE authority, then the exception CPF8A83 will be returned.

- Determine a user's level of authority to a document library object. (See Figure 6-2 on page 6-17 for an example of a program that checks authorization.)
- Check if a user has at least a specified authority to a document library object when working on behalf of another user.
CHKDLO DLO(object-name) FLR(folder-name) AUT(*ALL)
USRID(MARY SYSTEM1)

If the CHKDLO command completes without an exception, then Mary on system SYSTEM1 (and anyone who works on her behalf) has *ALL authority to the object.

Rules

The following rules apply to using the CHKDLO command:

- The only levels of authority that can be checked with this command are *ALL, *CHANGE, *USE, and *EXCLUDE.
- *SYSOBJNAM must be used to check documents that are not in a folder.
- If you do not specify OBJTYPE, a check is made for either a document or folder of the name specified in the DLO parameter.

Copy Document (CPYDOC) Command

The Copy Document (CPYDOC) command allows you to:

- Copy a document from one folder to another folder.
- Copy a document not in a folder into a folder.
- Make another copy of a document in the same folder.
- Replace an existing document with another document.

Security Considerations

If you are copying a document into an existing document, then you must have *CHANGE authority to the existing document. You do not need authority to the folder containing it.

If you are creating a new document from the original, then you must have *CHANGE authority to the folder that is to contain the new document.

You must have at least *USE authority to a document to copy it. You do not need authority to the folder containing the original document.

If you replace an existing document, its authorizations are unchanged. If you create a new document, its authorizations are the same as those of the original document, except the user creating it is the owner.

Some Ways to Use This Command

To copy a document that you own into another user's folder, use the following command:

```
CPYDOC FROMDOC(mydoc) FROMFLR(myflr) TODOC(yourdoc) TOFLR(yourflr)
```

Since REPLACE(*YES) was not specified, you will not accidentally replace an existing document named YOURDOC. You must have *CHANGE authority to YOURFLR to do the copy.

The following command makes a copy of a document that is not in a folder:

```
CPYDOC FROMDOC(*SYSOBJNAM) TODOC(newdoc) TOFLR(myflr)
      SYSOBJNAM(system-object-name)
```

Either user-defined names or system-object names can be used with this command. System-object names must be used for documents that are not in a folder.

The default for the REPLACE parameter is to not replace an existing document. This protects you from accidentally copying into an existing document. If you want to replace an existing document, specify REPLACE(*YES). You can specify REPLACE(*YES) even if the target document does not exist.

Word Processing Uses for the CPYDOC Command

This command can be called from a CL program to copy a user-defined shell document to create a new document. By using the CPYDOC command followed immediately by the EDTDOC command, the user can essentially create a new document without having the Create Document Details display appear. This technique allows a user to select an option from an application menu to create a new document, and be taken directly to the Edit display. For example, to ensure all created documents have certain keywords in the document details, use this technique with a different shell document for each keyword.

You may also use this command to create a shell document with the same name over and over again without having to first delete the document. For example, by specifying *YES for the REPLACE parameter, a standard form document named FORM is created for each customer response. That same document may then be printed using the PRTDOC command. For the next customer response, the document FORM can again be created by copying a shell document and using the REPLACE option.

The following Copy Document command copies the document SHELL.FRM, which resides in folder SHELLS, to the document FORM in folder CUSTOMER, replacing this document if it already exists.

```
CPYDOC FROMDOC(SHELL.FRM) FROMFLR(SHELLS) TODOC(FORM)
      TOFLR(CUSTOMER) REPLACE(*YES)
```

The following Copy Document command copies the document SHELL.RPT, which resides in folder SHELLS, to the document REPORT. When the TOFLR parameter is not specified, REPORT is stored in the same folder that the original document resides in (SHELLS), because the TOFLR parameter takes *FROMFLR as its default. If a document named REPORT already resides in folder SHELLS, it will not be replaced, because the REPLACE parameter takes *NO as its default.

```
CPYDOC FROMDOC(SHELL.RPT) FROMFLR(SHELLS) TODOC(REPORT)
```

The following Copy Document command copies the document SHELL.LET, which resides in folder SHELLS, to the folder JONES. Since the TODOC parameter is not specified, the new document will also be named SHELL.LET, because the TODOC parameter takes *FROMDOC as its default. If a document named SHELL.LET already resides in folder JONES, it is not replaced, because the REPLACE parameter takes *NO as its default.

```
CPYDOC FROMDOC(SHELL.LET) FROMFLR(SHELLS) TOFLR(JONES)
```

The CPYDOC command copies all text from the document being copied to the new document. It also copies all format information, header and footer text, footnote pages, dictionary options, print options, and so on.

The CPYDOC command can be used to create a new document in a batch environment, using the system profile document as a base. The system profile document, SYSTEM, would be specified for the FROMDOC parameter. The profile folder, QPRFFLR, would be specified as the FROMFLR parameter.

Copy From PC Document (CPYFRMPCD) Command

The Copy From PC Document (CPYFRMPCD) command copies the data in a document to a system physical database file. When you use WRKDOC to display the documents in a folder, PC files are listed as documents with a type of PCFILE.

Note: This command is usually used to copy PC files that are in a folder to a database file. This command differs from RTVDOC in that CPYFRMPCD can convert some PC files into a format that is usable on the AS/400 system.

Security Considerations

You must have object operational authority to the CPYFRMPCD command. You must have at least *USE authority to the PC document.

You must have one of the following authorities to the physical file:

- *OBJOPR and *ADD to use the MBROPT(*ADD) parameter
- *OBJOPR, *OBJMGT, *ADD, and *DLT to use the MBROPR(*REPLACE) parameter

If a new member is created in the physical file, the following authorities are required:

- *OBJOPR, *OBJMGT, and *ADD to the file
- *OBJOPR, *READ, and *ADD to the library containing the file

Some Ways to Use This Command

You can use CPYFRMPCD to copy a PC text file into a source database member. You can then edit the member using STRSEU. For example, if you had created a PC ASCII text file on your PC and then saved the file in a folder, you could then convert the PC file into a fixed-length EBCDIC file. The command

```
CPYFRMPCD FROMFLR(myflr) FROMDOC(pcdoc) TOFILE(myfile/pcdoc)
```

would copy MYFLR/PCDOC to the first member in MYFILE/PCDOC.

Note: To create a source physical file, see the Create Source Physical File (CRTSRCPF) command. To create a physical data file, see the Create Physical File (CRTPF) command. These commands can be found in the *CL Reference* manual.

Rules

The following rules apply to this command:

- If the physical file is a *SOURCE file, then a sequence number and a system date of zeros are added to each record as it is copied to the file. The Override Database File (OVRDBF) command can be used to override the physical file name and the other parameters of this command.
- The PC document can either replace the contents of the physical file member or it can be added to the end of the member.

Hints and Techniques

To copy all the PC files in a folder to members in a database file, you could do the following:

- Use DSPFLR to get a list of documents in a folder.
- Test each document to see if it is of type PCFILE.
- Copy each document that passes the test to a member in the database file. Use PC file name without the extension as the name of the member.

You must specify the format of the records in the PC document. Use one of the following formats:

TRNFMT(*TEXT) Use this format when the records are in standard DOS ASCII variable-length format. This format is used by many PC text editors. The records are transformed to a fixed-length format. The carriage return (CR), line feed (LF), and end of file (EOF) characters are removed from the file. Embedded tab characters are expanded as blanks, and the record is padded with blanks to fill out the fixed-length record. If a record is too long to fit into the fixed-length format, an exception is created.

TRNFMT(*NOTEXT) Use this format when the records in the PC document are the same length as the database file to which they are being copied.

Character Set and Code Page Considerations

When translation is specified, the data in the document is translated from ASCII to EBCDIC characters. A translation table is used to perform the translation. This table consists of ASCII characters mapped to EBCDIC characters. You can use the system translation table by specifying TRNTBL(*DFT), an explicit table, or no table at all. If you specify TRNTBL(*NONE), no translation occurs. This is useful if you just want to make a copy of the PC document as is.

Copy To PC Document (CPYTOPCD) Command

The Copy To PC Document (CPYTOPCD) command copies a member of a system database file to a PC document. A PC document is any object in a folder that is in a PC format. When you use WRKDOC to display the documents in a folder, PC files are listed as documents with a type of PCFILE.

Security Considerations

You must have object operational authority to the CPYTOPCD command. You must have *OBJOPR and *READ authority to the physical file. You must have *READ authority to the library that contains the database file.

If the PC document that you are copying to exists, then you must have *CHANGE authority to the PC document. You do not need authority to the folder containing the PC document.

If the PC document that you are copying to does not exist, then you must have *CHANGE authority to the folder in which the PC document is to be created.

Some Ways to Use This Command

You can use CPYTOPCD to copy the data contained in member MBRNAME of file XYZ/MYDATA to a folder named YOURFLR. The PC document that contains the data is named NEWDATA.TXT. The data is translated from EBCDIC to ASCII using the default translation table. Records of all formats are copied. If the PC document already exists in the folder, the copy is not performed because the default for the REPLACE parameter is *NO.

```
CPYTOPCD FROMFILE(XYZ/MYDATA) FRMMBR(MBRNAME) TOFLR(YOURFLR)
TODOC(NEWDATA.TXT)
```

Rules

A valid document name must be used for the PC file that is to be copied to. If the PC file does not exist, it is created and owned by the user making the copy.

The database file you are copying from can be either a physical file or a logical file. If this file is a *SOURCE file and the records are being translated from EBCDIC to ASCII, the sequence number and date are removed when the records are written in the PC document. The Override Database File (OVRDBF) command can be used to override the file name and the other parameters of this command.

Hints and Techniques

The default for this command is to not replace an existing PC document. You can override this default by using REPLACE(*YES).

You can use the RCDFMT parameter to specify if all record formats in the file are copied or if only those records with a specific format are copied. RCDFMT(*ALL) specifies that all record formats should be copied. RCDFMT(*record-format-name*) specifies the name of the record format that should be copied.

You must specify the format of the records in the PC document. Use one of the following formats:

TRNFMT(*TEXT) Use this format when the records are to be in standard DOS ASCII variable-length format. This format is used by many PC text editors. The records are transformed to a variable-length format. A carriage return (CR) and line feed (LF) are added to the end of each line. An end of file (EOF) is added to the end of the file.

TRNFMT(*NOTEXT) Use this format when the records in the PC document should be the same length as the database file from which they are being copied.

Character Set and Code Page Considerations

When translation is specified, the data in the database is translated from EBCDIC to ASCII characters. A translation table is used to perform the translation. This table consists of EBCDIC characters mapped to ASCII characters. You can use the system translation table by specifying TRNTBL(*DFT), an explicit table, or no table at all. If you specify TRNTBL(*NONE), no translation occurs. This is useful when you make a copy of the database member as is.

Create Folder (CRTFLR) Command

The Create Folder (CRTFLR) command allows you to create a first-level folder or create a folder within an existing folder.

Note: To delete a folder, use the Delete Document Library Object (DLTDLO) command.

Security Considerations

If the folder is being created in an existing folder, you must have *CHANGE authority to the existing folder. Public authority can be specified when the folder is created. The authority can be either explicitly specified, specified to an authorization list, or defaulted to the previous-level folder.

Note: If you use an authorization list for the public authority, you also get the explicit authorizations in the authorization list.

If the AUT parameter is not specified, public authority takes the following default values:

- *EXCLUDE for first-level folders
- The level of authority of the previous-level folder

To restrict the use of this command for creating folders, change the public authority for the command to *EXCLUDE using the EDTOBJAUT command. Only security officers and explicitly authorized users can then create folders.

To control creation of first-level folders, but still allow users to create folders within folders, the authority to the root (*ROOT) folder must be changed.

Some Ways to Use This Command

To create a first-level folder called PAYROLL, use the following command:

```
CRTFLR FLR(PAYROLL) INFLR(*NONE)
```

The text description of this folder is PAYROLL, as it uses the folder name as the default. The public authority to this folder is *EXCLUDE.

To create a folder called "1987" within the PAYROLL folder:

```
CRTFLR FLR(1987) INFLR(PAYROLL) AUT(*CHANGE) TEXT('1987 payro11')
```

The public is granted *CHANGE authority to the folder, 1987, which allows them to add documents to the folder, change the folder description, or display the contents of the folder. The public authority to PAYROLL is still *EXCLUDE.

Rules

First-level folder names must be unique within the system. Within a folder, folder and document names must be unique. You can use the same folder name within multiple folders. For example:

```
MYFLR/PAYROLL/1987
MYFLR/PAYROLL/1988
YOURFLR/PAYROLL/1987
YOURFLR/PAYROLL/1988
```

The folder names in the example are valid. The PAYROLL folder in MYFLR is not the same folder as the PAYROLL folder in YOURFLR.

When creating folders, limit the number of apostrophes in the name to 2. The Merge Document (MRGDOC) command cannot process folder names with more than 2 apostrophes if submitted in batch. The number of subfolders should be limited to 4, if using apostrophes in the name.

Character Set and Code Page Considerations

The CMDCHRID parameter specifies the character identifier (graphic character set and code page) for the data being entered as command parameter values (applies to TEXT parameter).

CMDCHRID(*SYSVAL)

Specifies that the system graphic character set and code page should be used.

CMDCHRID(*DEVLD)

Specifies that the device character set and code page should be used. This option is valid only when specified from an interactive job. If specified in an interactive CL program or a batch job, an error occurs.

CMDCHRID(GRAPHIC-CHARACTER-SET CODE-PAGE)

Provides a specific graphic character set and code page to be used.

Delete Document Library Object (DLTDLO) Command

The Delete Document Library Object (DLTDLO) command allows you to delete documents and folders in the document library.

Security Considerations

To delete a document or folder, you must have either *ALL authority to the document or folder, *ALLOBJ special authority, or *SECADM special authority.

Only a user with *ALLOBJ or *SECADM special authority can specify DLO(*ALL) when using FLR(*ANY) or FLR(*NONE). Only a user with *ALLOBJ or *SECADM special authority can specify OWNER(*ALL) or specify a user profile name other than that belonging to the user who issues this command.

Some Ways to Use This Command

To delete an empty folder:

```
DLTDLO DLO(MYFLR)
```

To delete a folder that contains folders and documents:

```
DLTDLO DLO(*ALL) FLR(MYFLR)
```

Note: You must have *ALL authority to all the objects in the folder in order to delete them.

To delete all documents with a document class of LETTERS filed in the system during August 1987:

```
DLTDLO DLO(*SEARCH) CRTDATE((*AVAIL 080187) (*AVAIL 083187))  
CLS(LETTERS) OWNER(*ALL)
```

Note: Only a user with *ALLOBJ or *SECADM can use the OWNER(*ALL) parameter.

To delete all documents that belong to the user issuing the command:

```
DLTDLO DLO(*SEARCH) CRTDATE((*AVAIL *BEGIN) (*AVAIL *END))  
OWNER(*CURRENT)
```

Note: If you do not specify CRTDATE, only documents filed on the current date are deleted.

Rules

Once you press the Enter key, the specified objects are deleted. There is no prompting before the document or folder is deleted. If a document or folder is in use, the object is not deleted.

When more than one document or folder is specified, but a document or folder cannot be deleted because it is in use or the user is not authorized, a message is sent, and the function continues to delete those objects that remain in the list.

Hints and Techniques

You can use any combination of CRTDATE, CHKEXP, DOCCLS, CMDCHRID, FLR, and OWNER when the DLO is *SEARCH. When more than one of these parameters is used, the resulting list consists of those objects that meet all the conditions.

DLO(*SEARCH) can be used to find a set of document library objects that meet certain requirements and then delete those objects. The details that you can test are:

| | |
|----------------|------------------------------|
| CRTDATE | Creation date |
| CHKEXP | Expiration date |
| DOCCLS | Document class |
| OWNER | Object owner |
| FLR | Folder containing the object |

Note: You can use the CMDCHRID (character identifier) parameter with the DOCCLS parameter to specify what character set and code page the document class was stored in.

Display Document Library Object Name (DSPDLONAM) Command

The Display Document Library Object Name (DSPDLONAM) command accepts one form of a name for a document or folder object and converts the name to the other forms of the name for that object. The command then displays or prints the results.

Security Considerations

To use this command, you must have at least *USE authority to the document or folder or *ALLOBJ special authority. You must have *ALLOBJ special authority to use this command with a distribution document.

Some Ways to Use This Command

Use this command to display or print the various names of a filed document, folder, or distribution document when one form of the name is known. For example, given a document library object (DLO) name and folder path, document ID, LADN timestamp, or system object name, the DSPDLONAM command displays or prints the equivalent DLO name, folder path, document ID, system object name, and object class.

The following table shows the names returned for various object types.

| DSPDLONAM Description Specified | Values Returned for Object Types | | | | |
|---------------------------------|----------------------------------|---------------------------|--------------------------|--------------------|-------------|
| | Distribution Document | Folderless Filed Document | Filed Document in Folder | First-level Folder | Subfolder |
| Object class | *DST | *DOC | *DOC | *FLR | *FLR |
| Document library object | *NONE | *NONE | DLO name | DLO name | DLO name |
| Folder | *NONE | *NONE | Folder path | *NONE | Folder path |
| Document identifier | *NONE | Document ID | Document ID | Document ID | Document ID |
| LADN timestamp | Timestamp | Timestamp | Timestamp | Timestamp | Timestamp |
| System object name | Object name | Object name | Object name | Object name | Object name |

To find the document MYDOC in folder MYFLR and display the various forms of the document's name and the object class (*DOC), use the following command:

```
DSPDLONAM DLO(MYDOC) FLR(MYFLR)
```

To find the document specified by variable &DOC in folder path PAYROLL/SEATTLE and display the various forms of the document's name and its object class (*DOC), use the following command from a CL or REXX language program:

```
DSPDLONAM DLO(&DOC) FLR('PAYROLL/SEATTLE') OBJCLS(*DOC)
```

To find the folder with document identifier 1990121710463785SYSTEM2 (the folder was created on SYSTEM2 on December 27, 1990, at 10:46:37.85 a.m.) and display the various forms of the folder's name and its object class (*FLR), use the following command:

```
DSPDLONAM DLO(*DOCID) DOCID(1990122710463785SYSTEM2) OBJCLS(*FLR)
```

To find the distribution document with the LADN timestamp 07C60C1B0E2F112A and display the various forms of its name and its object class (*DST), use the following command:

```
DSPDLONAM DLO(*LADNTSP) LADNTSP(07C60C1B0E2F112A) OBJCLS(*DST)
```

You must have *ALLOBJ authority to use the command with a distribution document.

To find the document with the system name DS5S471742 and print the various forms of the document's name and its object class (*DOC), use the following command:

```
DSPDLONAM DLO(*SYSOBJNAM) SYSOBJNAM(DP5S471742) OUTPUT(*PRINT)
```

The printed output from the DSPDLONAM command has the following format:

```

                                Display DLO Name                                Page    1
5738SS1 V2R2M0  910517                                RCHAS753 04/18/91 13:55:54

Document library object. . . . . : MYDOC
      Folder . . . . . : MYFLR
Document identifier . . . . . : 1990122710463785SYSTEM2
LADN timestamp . . . . . : 07C60C1B0E2F112A
System object name . . . . . : DP5S471742
Object class . . . . . : *DOC

      * * * * *   E N D   O F   L I S T I N G   * * * * *
```

Rules

When this command runs, the system searches for the specified object. If the object is found, the system verifies that the user has *USE authority or *ALLOBJ special authority. If the object is not found or the user does not have sufficient authority, an escape message is sent.

If the folder that contains the specified document is in use or damaged, you cannot use the DSPDLONAM command with the folder path and DLO name to display the other forms of the document name. However, you can use the command with the document identifier, system object name, or LADN timestamp.

Display Folder (DSPFLR) Command

The Display Folder (DSPFLR) command can be used to display information, print information, or create a database file (OUTFILE) with information about the documents within a folder or about the folders that are within a folder.

Security Considerations

Any user can use this command. When displaying a list of first-level folders, only those that the user is authorized to are listed. When displaying a list of objects within a folder, all objects in the folder are listed, but objects that the user is not authorized to are marked *NOT AUTHORIZED.

Some Ways to Use This Command

List all the first-level folders that you are authorized to:

```
DSPFLR FLR(*ALL)
```

See all the documents in a folder:

```
DSPFLR FLR(folder-name) TYPE(*DOC)
```

See all the folders in a folder:

```
DSPFLR FLR(folder-name) TYPE(*FLR)
```

Put a list of all the folders and documents in a folder into a database file. All the levels of folders within the folder are listed.

```
DSPFLR FLR(folder-name) OUTPUT(*OUTFILE) LEVEL(*ALL)
OUTFILE(library/outfile-name) OUTMBR(member-name *REPLACE)
```

Note: LEVEL(*ALL) can only be used with OUTPUT(*OUTFILE) or OUTPUT(*PRINT).

Rules

If the output file does not exist it is created with a public authority of *EXCLUDE. If the output file member does not exist it is created with the same name as the output file. If an output file member is not specified, the first member in the output file is used. If OUTMBR(MEMBER-NAME *REPLACE) is used, the member is cleared first before the information is added. If OUTMBR(MEMBER-NAME *ADD) is used, the information is added to the end of the member.

Output File Layout for DSPFLR

The model output file used when DSPFLR TYPE(*FLR) OUTPUT(*OUTFILE) is specified is named QADSPFLR in library QSYS. The record format name is FLRDTL. Refer to Appendix C, "Office Services Output File Considerations," for the output file layout for this version of the command.

File Document (FILDOC) Command

The File Document (FILDOC) command files information from a database file, from a mail item, or from command parameters to a document or as an Interchange Document Profile (IDP) for a document in the document library. (The IDP does not exist by itself in the document library.)

For TYPE(*FILE):

- The data for the document is supplied from a database file. The IDP can be supplied from either a database file or specific IDP parameters. If both IDPFILE and one or more IDP command parameters are specified in the command, IDPFILE takes precedence and the command parameters are ignored.

Note: IDP parameters are also referred to as document details or document attributes.

The database file can be any of the following:

- A formatted output file from the RCVDST or RTVDOC commands
- A formatted file similar to the output file from RCVDST or RTVDOC commands
- A user-defined output file

If the output file is from RCVDST or RTVDOC (or formatted the same) then when the document is filed, the data is recognized in this format and properly processed to format the data and the IDP. You can access this document by using the word processor.

If the output file is not in the RCVDST or RTVDOC format, then the data is stored in the document. The type becomes OTHER. You cannot access this document through the word processor, but it acts as a storage place for your data. You can convert your data back to a database file by doing a RTVDOC, but the data will be in the RTVDOC format in record code 800.

For TYPE(*IDP):

- The IDP can either be specified by parameters on the command or be located in a database file. The IDP command parameters are ignored if IDPFILE is specified. If you specify the IDP with parameters, the document is a HARD-COPY document (the file cabinet location (FILCAB) parameter must be specified). Otherwise, when the IDP is specified as located in a database file (IDPFILE parameter), the document has only an IDP and no data.
- You cannot replace the IDP in an existing document using the FILDOC command. To change the IDP in an existing document, use the CHGDOCD command.

For TYPE(*DSTID):

- The IDP parameters can be taken from the distribution document, from the IDP command parameters, or from a database file. The IDP command parameters are ignored if IDPFILE is specified.
- You can file a document directly from the mail log into the document library by specifying the distribution ID.
- If you specify IDPFILE(*DSTIDIDP), the IDP information associated with the distribution document is used. If you specify IDP values on the FILDOC, then those are used for the IDP of the filed document.
- The distribution ID extension takes 01 as its default. If multiple copies of a distribution are sent to you, and you want to file them with FILDOC, you need to specify the distribution extension. The distribution extension can be found by doing a QRYDST for incoming distributions to an output file and, at offset 144-145, the distribution extension can be found.
- The document type, document CHRID, and system code are the same as those associated with the distribution.

Security Considerations

The user of this command must have *CHANGE authority to the folder, *ALLOBJ special authority, or authority to work on behalf of the specified user ID address. The specified user ID must also have *CHANGE authority to the folder to file it. Use the GRTUSRPMN command to get permission to work on behalf of another user.

Some Ways to Use This Command

To file the distributions that you receive, use a CL program by doing a QRYDST of the incoming distributions to an output file. For each distribution, do a FILDOC of the distribution ID and distribution ID extension to a file in a specified folder.

You may have done a RCV DST or RTV DOC (without checking the document out) and changed the file with your own application or performed some file functions with it. Now you want to file it back into a document. You can use the FILDOC command and file it into a new document (as long as any changes are in a recognizable format).

Rules

If the document already exists in the folder, then it is not replaced. The document must not already exist in the folder in order for the file to be successful.

If the IDP is coming from a file, it comes from record code 500. This record can be created from the RCVDST or RTVDOC command with the parameter OUTDTATYP(*IDP) or OUTDTATYP(*ALL) specified, so the file contains record type 500. This record can be user-created if the record follows the IDP format specified by DIA.

The document type must accurately reflect the actual type of data. The data is not checked.

Each document has a document identifier. If the document is without a folder, then the document identifier is used in the DOCID parameter (for example, RTVDOC).

FILCAB must be specified if TYPE(*IDP) and IDPFILE(*NONE) are specified. This makes the document a HARDCOPY document.

Revisable-Form Text: Document Content Architecture Considerations

The document text type of the document created by the FILDOC command is revisable-form text (RFT). Print options, merge options, and user dictionaries are not supported in this organization. See the *Revisable-Form Text Reference* manual for additional information.

Print options are derived hierarchically from the active text profile, the system profile, and the user profile.

Merge options and user dictionaries are not carried forward.

Hints and Techniques

If you want to edit the document after filing it, you must specify ALWRPL(*YES). The default value for this field is ALWRPL(*NO) which specifies this as a final document that cannot be changed.

If a System/38 document exists in a database file and you want to file it so it can be accessed through OfficeVision/400, you must specify DOCTYPE(65262) and SYSCODE('IBM SYSTEM/38').

Character Set and Code Page Considerations

The USRID parameter is translated from the value specified in the CMDCHRID parameter to character set 930, code page 500.

The following parameters are stored in the document without translation and are considered to be in the character set and code page specified in the CMDCHRID parameter. They are also stored in the document and folder search index. If possible, they are translated to character set 697, code page 500 before being stored in the index. Otherwise, they are stored in the database without being translated and are considered to be in the character set and code page specified in the CMDCHRID parameter.

- SYSCOD
- DOCD

- AUTHOR

- DOCCLS
- KWD
- SUBJECT
- FILCAB
- CPYLST
- REFERENCE
- STATUS
- PROJECT

Move Document (MOVDOC) Command

The Move Document (MOVDOC) command allows you to:

- Move a document from its current folder to another existing folder.
- Move a folderless document into a folder.
- Move a document that is in a folder out of the folder.

Note: Documents that are not in a folder but are in the document library are referred to as **folderless documents**. You can use the document search function of OfficeVision/400, or the QRYDOCLIB command to find folderless documents. They can only be referred to in this command by their ten-character, system-object name.

Security Considerations

You must have *CHANGE authority to both the folder from which you are moving the document and the folder to which you are moving the document. You must have *ALL authority to the document you are moving, regardless of where you are moving it.

All authorities to the document are kept as is; they are not changed as a result of the move.

Some Ways to Use This Command

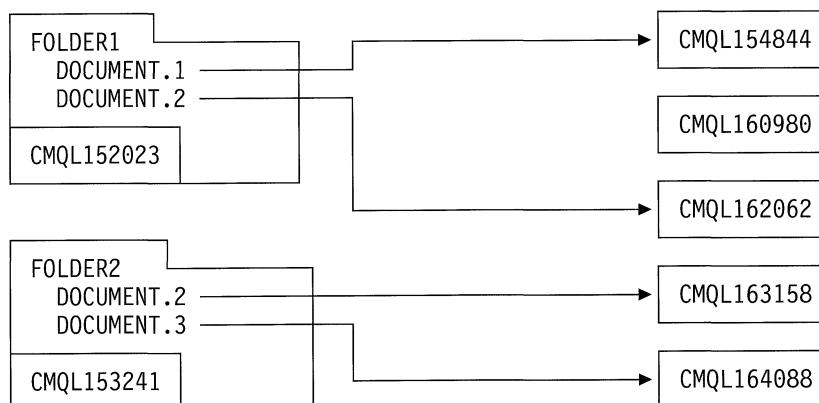


Figure 5-4. Folders and Documents Before Moving

Figure 5-4 shows the following objects:

- CMQL152023** A folder named FOLDER1
- CMQL153241** A folder named FOLDER2
- CMQL154844** A document that is in FOLDER1 and has the name DOCUMENT.1

- CMQL160980** A document that is not in any folder and so does not have a user-defined name
- CMQL162062** A document that is in FOLDER1 and has the name DOCUMENT.2
- CMQL163158** A document that is in FOLDER2 and has the name DOCUMENT.2
- CMQL164088** A document that is in FOLDER2 and has the name DOCUMENT.3

You can move DOCUMENT.1 from FOLDER1 to FOLDER2 using the following command:

```
MOVDOC FROMFLR(FOLDER1) FROMDOC(DOCUMENT.1) TOFLR(FOLDER2)
```

Since one document (DOCUMENT.2) appears in both folders (FOLDER1 and FOLDER2), you cannot move DOCUMENT2 from FOLDER1 to FOLDER2 unless you do one of the following:

- Delete DOCUMENT.2 from FOLDER2 before you move it.
- Rename DOCUMENT.2 in FOLDER2 before you move it.
- Rename DOCUMENT.2 as you move it from FOLDER1 to FOLDER2. For example, to rename DOCUMENT.2 to DOCUMENT.4 while moving it from FOLDER1 to FOLDER2:

```
MOVDOC FROMFLR(FOLDER1) FROMDOC(DOCUMENT.2)
TOFLR(FOLDER2) RENAME(DOCUMENT.4)
```

You can move the folderless document, CMQL160980, into FOLDER1 and assign it the name DOCUMENT.5 using the following command:

```
MOVDOC FROMDOC(*SYSOBJNAM) SYSOBJNAM(CMQL160980)
TOFLR(FOLDER1) RENAME(DOCUMENT.5)
```

You can move DOCUMENT.3 from FOLDER2, making it a folderless document using the following command:

```
MOVDOC FROMFLR(FOLDER2) FROMDOC(DOCUMENT.3) TOFLR(*NONE)
```

After the folders and documents have been moved, the folder paths to the documents will look like those in Figure 5-5.

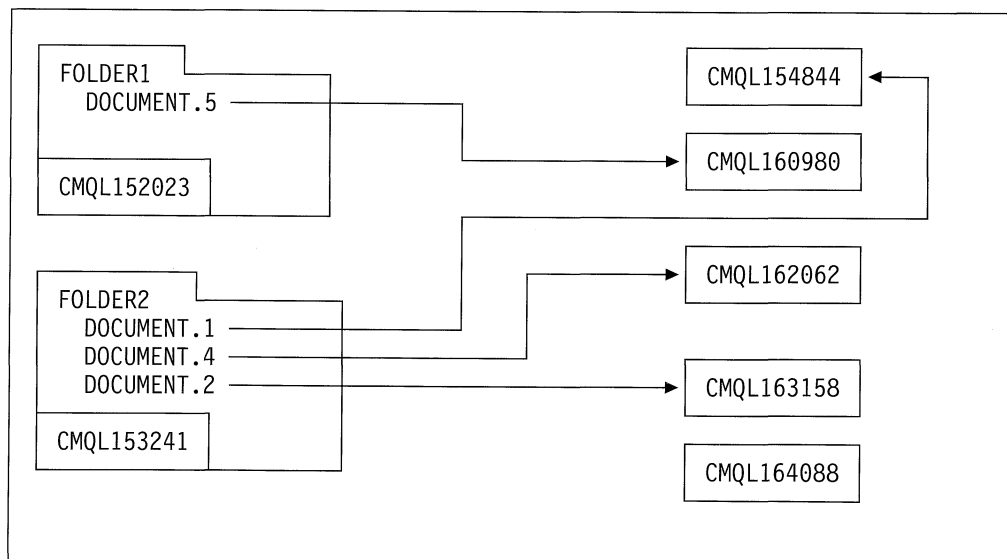


Figure 5-5. Folders and Documents After Moving

Rules

The system-object name of the document cannot be changed with the MOVDOC command.

The MOVDOC command cannot replace an object that is already in a folder.

This command does not create or delete objects. It only changes the folder path to a document.

If you know the system-object name, you can use that name even if the document has a user-defined name.

Hints and Techniques

You have the following options if the document you are moving has the same name as a document presently in the folder you are moving it to:

- Use the Move Document (MOVDOC) command to rename the document as you are moving it.
- Use the Rename Document Library Object (RNMDLO) command to rename the document in the folder that you are moving a document to, and then use the MOVDOC command.
- Use the Delete Document Library Object (DLTDLO) command to delete the document that is in the folder you are moving a document to, and then use the MOVDOC command.
- Use the Copy Document (CPYDOC) command to replace the document in the folder you are moving it to and then use the DLTDLO command to delete the document you want to move.

Remember that the authorities of the document are not changed as a result of using the MOVDOC command to move the document to another folder. For example, you have a document in FOLDER3 and a document in FOLDER4 that both have the name DOCUMENT.6. You want to move DOCUMENT.6 from FOLDER3 to FOLDER4.

- You may want to keep the authorities, system-object name, and library-assigned document name (LADN) of the document that you are moving.
Use the DLTDLO command to delete DOCUMENT.6 from FOLDER4 or use the RNMDLO command to rename DOCUMENT.6 in FOLDER4. Or you can use the MOVDOC command to move the document in FOLDER3 to FOLDER4 and at the same time rename the document. The authorities, system-object name, and LADN of DOCUMENT.6 in FOLDER3 are not changed as a result of moving the document to FOLDER4.
- You may want to keep the authorities, system-object name, and LADN of DOCUMENT.6 in FOLDER4 but replace that document's contents with the contents of DOCUMENT.6 in FOLDER3.
Use the CPYDOC command to replace DOCUMENT.6 in FOLDER4 with the contents of DOCUMENT.6 in FOLDER3. Then use the DLTDLO command to delete DOCUMENT.6 from FOLDER3. The document authorities, system-object name, and LADN of DOCUMENT.6 in FOLDER4 are not changed as a result of the copy.

Query Document Library (QRYDOCLIB) Command

The Query Document Library (QRYDOCLIB) command allows a user to search for documents that are filed in the document library and to copy information about the documents that satisfies the search request into an output file, a document list, or both.

The output file can be processed by a program to perform operations on the documents that met the search values or to produce a customized report.

A document list is a document that contains references to documents in the document library that met the search criteria.

The document list or outfile captures a view of the document library at the time the QRYDOCLIB command was run. The QRYDOCLIB command must be run again to update the document list or the outfile.

Search information can include document details (using the QRYDFN parameter), document text (using the QRYTXT parameter), or both. Document detail search includes such information as keywords, author, and description. Document text search locates documents according to the search on the text of a document. Phrases, words, or word parts, or a combination of these can be searched on as part of the text search. For more information, see Chapter 7, "Text Search Services."

The document lists created by the QRYDOCLIB command can be used in OfficeVision/400 by selecting the Documents and folders option (option 5 on the OfficeVision/400 menu) and then the Work with document lists option (option 4 on the Documents and Folders menu). All options on the Work with Owned Document Lists display, such as, option 5 (Work with documents in list), can be used on a document list created with the QRYDOCLIB command except for the following:

- Option 2 (Change criteria)
- Option 7 (Copy criteria)
- Option 8 (View criteria)
- Option 9 (Run search)

To change the search values or to run the search again, the QRYDOCLIB command must be changed and run again to update a document list created by the QRYDOCLIB command.

Security Considerations

The user must have at least *USE authority to each document selected for inclusion in the document list to use QRYDOCLIB. Objects found that the user does not have at least *USE authority to do not appear in the list. If the current user of this command issues this command on behalf of another user, he must have *ALLOBJ special authority or the authority to work on behalf of the other user given to him by using the GRTUSRPMN command.

Some Ways to Use This Command

You can use the QRYDOCLIB command to find all documents that meet certain requirements and put the result to an output file. You can then use this file in a CL program to process each document.

If you have specific document classes that you use and you want to enforce them, you can run the QRYDOCLIB command for all document classes that do not match your list. Process those documents and send a message to the owner of those documents (using the &QDLUID and &QDLADR fields) requesting they be changed using the CHGDOCD command, or change them yourself. The following QRYDOCLIB command returns all documents that do not match the document classes LETTER, HARDCOPY, or DOCUMENT:

```
QRYDOCLIB QRYDFN(*IF ((*DOCCLS *NE LETTER *AND) (*DOCCLS *NE HARDCOPY)
(*DOCCLS *NE DOCUMENT))) OUTFILE(library-name/file-name)
OUTDTATYP(*DOCCLS)
```

You can also create an output file to check all of the documents that have not yet been completed, but which have an action date (due date) between two specified dates. The following command returns all documents that have a due date that falls between the specified dates:

```
QRYDOCLIB QRYDFN(*IF ((*ACTDATE *GE &BDATE *AND) (*ACTDATE *LE &EDATE)))
OUTFILE(QTEMP/QRYDOC) OUTDTATYP(*ACTDATE *CMPDATE)
ORDER(*ACTDATE)
```

You can process the output file to see if the document has a completion date. You can then send a message to inform the user of a document having an action date. Figure 5-7 on page 5-45 shows a sample program that uses the output file you create with the QRYDOCLIB command.

You can set up your search values to exclude documents that might otherwise be found during a search. If you want to create a document list of all documents that contain the word *report* but not the word *annual*, use the following command:

```
QRYDOCLIB QRYTXT(*IF (('Report' *ALL *NO *ANDNOT) ('Annual')))
OUTFILE(library-name/file-name)
```

You can use the QRYDOCLIB command to search all the subfolders of a specified folder. However, only 100 folders are allowed as input to the command. The subfolders within the folder count toward the 100 folder limit. If you specify that you want to search subfolders, you can specify only one folder name to prevent you from exceeding the 100 folder limit.

Note: Specifying one folder name prevents you from exceeding the 100 folder limit only if the folder contains 99 or fewer folders (folders within folders within folders, and so on, are counted).

Use the following command to search a folder and all its subfolders:

```
QRYDOCLIB FLR(folder-name) SCHSUBFLR(*YES) OUTFILE(library-name/file-name)
```

Rules

The output file format must be the same as OSQDL of the system file QSYS/QAOSIQDL. If the output file does not exist at the time the QRYDOCLIB command is run, the command creates the output file with the correct format.

Note: When using selection values for the query, the AND operator is used to group conditions, not the OR operator.

Refer to Appendix C, “Office Services Output File Considerations,” for more information on the output file layout for this command. See the *IF option of the QRYDFN parameter and the *IF option of the QRYTXT parameter for the QRYDOCLIB command in the *CL Reference* manual for more detailed information on specifying search values.

Hints and Techniques

Use the following hints and techniques for the QRYDOCLIB command:

- If you want the documents back in ascending or descending order by a particular field(s), you can specify this with the ORDER parameter. The performance may be affected when this is done.
- When multiple QRYDOCLIB commands are run in batch, the document list name and output file name must be different or an error occurs.
- You can find blank fields on all profile parameters.
- If you are using the QRYDOCLIB command to create a document list, a limit of 32 767 records can be copied into the list. If you are creating an outfile, there is no limit to the number of records.

Character Set and Code Page Considerations

The USRID parameters are translated from the value specified in the CMDCHRID parameter to character set 930 code page 500.

QRYDFN(*OWNER) is translated to character set 930 and code page 500. The values specified for QRYDFN are translated so that the character set and code page are common for the search database. If possible, the following parameters are translated to character set 697 and code page 500 before being stored in the database:

- DOCL
- TEXT
- QRYDFN

Rename Document Library Object (RNMDLO) Command

The Rename Document Library Object (RNMDLO) command changes the user-defined name of a document or folder. The system-object name cannot be changed.

You must have *ALL authority to the document or folder in order to rename it. You must have *CHANGE authority to the folder containing the object being renamed.

The document or folder being renamed cannot be in use when the RNMDLO command is run. When renaming a document, you need to specify a value for the FLR parameter.

Replace Document (RPLDOC) Command

The Replace Document (RPLDOC) command replaces content or IDP information in a document. This information comes from a database file that is output from the RTVDOC command. The information can also be a database file that is user-defined or output from the RCV DST command.

Only the document content record (record type 800) and the IDP details record (record type 500) can be read from the output file created by the RTVDOC and RCV DST commands. The user can specify to replace the document content, the IDP, or both. If the database file is user-defined, the document content or IDP is replaced in the document and, if the document content was changed, the document type becomes OTHER.

The document must be checked out before it can be replaced. To check out a document, specify CHKOUT(*YES) on the RTVDOC command. When a document is checked out it means that no one can update that document until it is checked back in. When the document is checked back in, the document can be replaced with new or changed content or IPD.

There is the option to specify either the document and folder name or the library-assigned document name (LADN). To specify a LADN, use the following:

```
TODOC(*DOCID) DOCID(library-assigned-document-name)
```

The LADN is a name assigned by the system when it is created. The LADN can be found by using the QRYDOCLIB command. It is also returned in the completion message when a FILDOC is done.

Security Considerations

If USRID(*CURRENT) is specified, the user of this command must have at least *CHANGE authority to the document being replaced or have *ALLOBJ special authority. If USRID (*user ID* and *address*) is specified, the user must have permission to work on behalf of the user specified, that user must have at least *CHANGE authority to the document or have *ALLOBJ special authority, and the document must not be marked personal.

Some Ways to Use This Command

The RPLDOC command is intended to be used with the RTVDOC command. Use the RTVDOC command to retrieve a document to an output file and check it out (specify CHKOUT(*YES)). Use your own applications to update the content or IDP records in the output file. Then use the RPLDOC command to insert the changed or new content or IDP into the document again. If you do not specify CHKOUT(*YES) on the RTVDOC command, the document is not checked out and can be changed by anyone with authority to do so. If you try to check it back in using the RPLDOC command, you are not able to do so because you did not check it out. Checking out a document guarantees the document cannot be changed by someone else.

You can replace a document with your own content or IDP by doing the following:

- Check out the document you want to replace:

```
RTVDOC FROMDOC(document-name) FROMFLR(folder-name)  
CHKOUT(*YES)
```

The document is then checked out without copying the contents to an output file.

- Use the RPLDOC command on the document just checked out with your user-defined database file. This copies your content or IDP to the document, stores it there, and, if you replace the document content, makes the type of the document OTHER.

To replace a document with text from another document, use the CPYDOC command with REPLACE(*YES) specified.

Rules

The document must be checked out before using this command. When a document is replaced with RPLDOC, the document is checked back in and can be updated by anyone who is authorized to it.

The document that is being replaced must exist.

Character Set and Code Page Considerations

The USRID parameter is translated from the value specified in the CMDCHRID parameter to character set 930 code page 500.

Retrieve Document Library Object Name (RTVDLONAM) Command

The Retrieve Document Library Object Name (RTVDLONAM) command accepts one form of name for a document or folder object and returns into program variables other forms of the name for that object. You can specify the DLO name and folder path, document identifier, LADN timestamp, or system object name.

Security Considerations

To use this command, you must have at least *USE authority or *ALLOBJ special authority. You must have *ALLOBJ special authority to use this command with a distribution document.

Some Ways to Use This Command

Use this command from a CL or REXX program to determine one name for a filed document, folder, or distribution document when another name is known. The following table shows the names returned for various object types.

| RTVDLONAM Parameter | Values Returned for Object Types | | | | |
|----------------------------------|----------------------------------|---------------------------|--------------------------|--------------------|-------------|
| | Distribution Document | Folderless Filed Document | Filed Document in Folder | First-level Folder | Subfolder |
| RTNOBJCLS (Object class) | *DST | *DOC | *DOC | *FLR | *FLR |
| RTNDLO (Document library object) | *NONE | *NONE | DLO name | DLO name | DLO name |
| RTNFLR (Folder) | *NONE | *NONE | Folder path | *NONE | Folder path |
| RTNDOCID (Document identifier) | *NONE | Document ID | Document ID | Document ID | Document ID |
| RTNLADNTSP (LADN timestamp) | Timestamp | Timestamp | Timestamp | Timestamp | Timestamp |
| RTNOBJNAM (System object name) | Object name | Object name | Object name | Object name | Object name |

To find the document MYDOC in folder MYFLR and return its document identifier in the variable &DOCID, use the following command:

```
RTVDLONAM DLO(MYDOC) FLR(MYFLR) RTNDOCID(&DOCID)
```

To find the document specified by the variable &DOC in folder path PAYROLL/SEATTLE and return its document identifier in variable &DOCID, use the following command:

```
RTVDLONAM DLO(&DOC) FLR('PAYROLL/SEATTLE') OBJCLS(*DOC)
RTNDOCID(&DOCID)
```

To find the folder with the document identifier 1990122710463785SYSTEM2 (the folder was created on SYSTEM2 on December 27, 1990, at 10:46:37.85 a.m.) and return the folder's system object name in the variable &OBJECT, use the following command:

```
RTVDLONAM DLO(*DOCID) DOCID(1990122710463785SYSTEM2) OBJCLS(*FLR)
RTNOBJNAM(&OBJECT)
```

To find the distribution document with the LADN timestamp 07C60C1B0E2F112A and return its system object name in variable &OBJECT, use the following command:

```
RTVDLONAM DLO(*LADNTSP) LADNTSP(07C60C1B0E2F112A) OBJCLS(*DST)
RTNOBJNAM(&OBJECT)
```

You must have *ALLOBJ authority to use this command with a distribution document.

To find the filed document with the system object name DP5S471742 and return the document's user-assigned name in variable &DOC and its folder path in variable &FLR (&DOC and &FLR will be *NONE if the document is folderless), use the following command:

```
RTVDLONAM DLO(*SYSOBJNAM) SYSOBJNAM(DP5S471742) OBJCLS(*DOC)
RTNDLO(&DOC) RTNFLR(&FLR)
```

Rules

When this command runs, the system searches for the specified object. If the object is found, the system verifies that the user has at least *USE authority to it. If the object is not found or the user does not have sufficient authority, an escape message is sent.

If the folder that contains the specified document is in use or damaged, you cannot use the RTVDLONAM command with the folder path and DLO name to retrieve the other forms of the document name. However, you can use the command with the document identifier, system object name, or LADN timestamp.

Retrieve Document (RTVDOC) Command

The Retrieve Document (RTVDOC) command allows you to copy information from a specific document to a database file, or to check out the document from the document library.

When a document is checked out, no one can update that document until it is checked back in. When the document is checked back in, it can be replaced with new or changed content or IDP. The checkout status can be reset with EDTDLOAUT or CHGDOCD commands. Any viewing, copying, or printing of a checked out document uses the document content as it was at the time the document was checked out.

Security Considerations

If USRID (*CURRENT) is specified, the user of this command must have at least *USE authority to the document being retrieved or have *ALLOBJ special authority. If USRID (user ID address) is specified, the user must have permission to work on behalf of the user specified, that user must have at least *USE authority to the document or have *ALLOBJ special authority, and the document must not be marked personal. If you want to check out the document, you must have *CHANGE authority to it.

Some Ways to Use This Command

You can use the RTVDOC command to send your data to an output file and check it out. Use the RPLDOC command to replace it in the document. If you want to get the document data into an output file to work with it and not replace the document, do not check it out.

If for some reason you do not want to have a document or a group of documents updated, you can use the RTVDOC command to check them out without creating an output file. Use RTVDOC FROMDOC(document-name) FROMFLR(folder-name) CHKDOC(*YES) to do this. To reset the checkout key, you can do it interactively from the EDTDLOAUT command or by the CHGDOCD command (batch or interactive).

Note: The instruction length may change when using the RTVDOC command to receive an existing document.

Rules

The document can only be checked in by the user who checked it out. However, the checkout status can be reset by any user who has *ALL authority to the document, *ALLOBJ special authority, or is the owner of the document (using the EDTDLOAUT command). The checkout status can also be reset by the owner or a user who has permission to work on behalf of the owner of the checked out document by using the CHGDOCD command.

If the output file specified already exists, it needs to be in the system format of QSYS/QAOSIRTV with record format OSRTVD. If the output file does not exist, then the RTVDOC command creates the output file with this format.

The document cannot be retrieved if it has been suspended and has been saved with STG(*FREE) specified on the SAVDLO command.

Output File Layout for RTVDOC

The model output file supplied with the system is named QAOSIRTV in library QSYS record format OSRTVD. Refer to Appendix C, "Office Services Output File Considerations," for more information about the output file layout.

The record codes in Figure 5-6 are written to the output file, depending on the options taken for the OUTDTATYP parameter.

Figure 5-6. Record Codes for RTVDOC

| Record Code | Description | OUTDTATYP Value | RTVCRS & RTVCPG Define Data |
|-------------|--|-----------------|-----------------------------|
| 105 | Document Description record | *DOCD *DFT *ALL | Yes |
| 110 | Creation Date and time | *CRTDATE *ALL | No |
| 115 | Expiration Date record | *EXPDATE *ALL | No |
| 120 | Document Date record | *DOCDATE *ALL | No |
| 125 | File Date record | *FILDATE *ALL | No |
| 130 | Date Last Changed record | *CHGDATE *ALL | No |
| 135 | Action Due Date record | *ACTDATE *ALL | No |
| 140 | Completion Date record | *CMPDATE *ALL | No |
| 145 | Author records | *AUTHOR *ALL | Yes |
| 150 | Copy List records | *CPYLST *ALL | Yes |
| 155 | Document Class record | *DOCCLS *ALL | Yes |
| 160 | File Cabinet Reference record | *FILCAB *ALL | Yes |
| 165 | Subject records | *SUBJECT *ALL | Yes |
| 170 | Keyword records | *KWD *ALL | Yes |
| 175 | Reference record | *REF *ALL | Yes |
| 180 | Status record | *STATUS *ALL | Yes |
| 185 | Project record | *PROJECT *ALL | Yes |
| 190 | Last Indexed Date and Time | *IDXDATE *ALL | No |
| 195 | Revision Date and Time | *OCDATE *ALL | No |
| 500 | Interchange Document Profile (IDP) records | *IDP *ALL | No |
| 800 | Document Data records | *DOC *DFT *ALL | Yes |

The OUTDTATYP value column tells what values for the OUTDTATYP parameter create that particular record. The records are created only if that data exists for the record code specified. For example, if Author records are requested and there is no data in the IDP for Author records, then type 145 or Author records in the output file are not created.

The *RTVCRS & RTVCPG Define Data* column tells whether the RTVCRS & RTVCPG fields in the record define the data in character set and code page for the data field RTVDTA.

For more detailed information about these record codes in the output file, see Appendix C, "Office Services Output File Considerations."

For more information about record code 500 (IDP record), refer to the *Interchange Document Profile Reference* manual.

For record type 800, the field RTVDTA contains the content of the document. The content is stacked in this 500-byte data field. The records contain 500 bytes, except for the last one, which may be less than 500.

Before the document content records are copied to the database file (record code 800), the content is converted into Revisable-Form Text:Document Content Architecture format (if necessary). Refer to Chapter 8, "Word Processing Services" for a list of document constructs that cannot be converted into Revisable-Form Text:Document Content Architecture.

Character Set and Code Page Considerations

The USRID parameter is translated from the value specified in the CMDCHRID parameter to character set 930 code page 500.

Note: If you attempt to display a file into which a document has been retrieved, translation does not occur from the character set and code page where the content is stored to the graphic character-set ID specified for the device.

Example Applications

The following program is an example for finding documents with an action date but no completion date:

```
PGM          PARM(&BDATE &EDATE &USRID &ADDR)
/* This program will do a QRYDOCLIB on all documents */
/* that have an action date greater than or equal to */
/* the beginning date (&BDATE) and less than or     */
/* equal to the ending date (&EDATE).                */
/*                                                    */
/* Output is a message to a specified user          */
/* (&USRID &ADDR) of documents that need action.    */
/*                                                    */
/* Example call of this program from the command    */
/* line would be:                                   */
/* CALL ACTDOC PARM('04/16/89' '04/22/89' MGR1 SYS1) */
/*                                                    */
/* where 04/16/89 is the beginning date and        */
/* 04/22/89 is the ending date. MGR1 is the        */
/* user ID at address SYS1.                         */
/*                                                    */
/* To compile this program use the CRTCLPGM command. */
/*                                                    */
DCL          VAR(&BDATE) TYPE(*CHAR) LEN(8) /* Variable +
           for the beginning date mm/dd/yy      */
DCL          VAR(&EDATE) TYPE(*CHAR) LEN(8) /* Variable +
           for the ending date mm/dd/yy        */
DCL          VAR(&USRID) TYPE(*CHAR) /* Variable +
           for the User ID parameter           */
DCL          VAR(&ADDR) TYPE(*CHAR) /* Variable +
           for the Address parameter           */
DCL          VAR(&SAVDUED) TYPE(*CHAR) LEN(8) /* +
           Save variable for the due date      */
DCL          VAR(&SAVDNM) TYPE(*CHAR) LEN(12) /* +
           Save variable for Document name     */
DCL          VAR(&SAVFLR) TYPE(*CHAR) LEN(63) /* +
           Save variable for Folder name       */
DCL          VAR(&MSG) TYPE(*CHAR) LEN(256) /* +
           Message data                       */
DCLF        FILE(QSYS/QAOSIQDL) RCDFMT(OSQDL) /* File +
           containing query of document library */

/* Query the document library for all documents */
/* that meet the requirements for &BDATE <= *ACTDATE */
/* <= &EDATE.                                     */
QRYDOCLIB QRYDFN(*IF ((*ACTDATE *GE &BDATE *AND) +
(*ACTDATE *LE &EDATE))) +
OUTFILE(QTEMP/QRYDOC) +
OUTDTATYP(*ACTDATE *CMPDATE) +
ORDER(*ACTDATE)
```

Figure 5-7 (Part 1 of 3). Program to Find Documents with an Action Date

```

/* Override the system database file for the QRYDOCLIB */
/* outfile. Change the file to the outfile from the */
/* QRYDOCLIB CL command. */
OVRDBF      FILE(QAOSIQDL) TOFILE(QTEMP/QRYDOC)

/* Now open the QRYDOCLIB outfile */
OPNDBF      FILE(QSYS/QAOSIQDL) OPTION(*INP)

/* Read all the records from the QRYDOCLIB outfile */
/* until end of file is reached */
READF:      RCVF          RCDFMT(OSQDL)

/* Monitor for EOF message */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* This record has the action date in the data. */
/* The records are ordered by record code and the */
/* only two records are action date and completion */
/* date (if there was a completion date). */

/* Save the due date (action date) */
DUERCD:     CHGVAR VAR(&SAVDUED) VALUE(%SST(&QDLDTA 5 2) *CAT +
          '/' *CAT %SST(&QDLDTA 7 2) *CAT '/' *CAT +
          %SST(&QDLDTA 3 2) )
/* Save the document name */
CHGVAR VAR(&SAVDNM) VALUE(&QDLDNM)
/* Save the folder name */
CHGVAR VAR(&SAVFLR) VALUE(&QDLFLR)

/* Read the next record and if this record is a */
/* completion date record (record code = 140) then */
/* this document is already handled, otherwise */
/* send a message to the &USRID &ADDR indicating */
/* this document needs action. */
RCVF          RCDFMT(OSQDL)

/* Monitor for EOF message */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* If this record is a completion date record then */
/* this document does not need action. Read the */
/* next record. */
IF          COND(&QDLRC *EQ 140) +
          THEN(GOTO CMDLBL(READF))
ELSE +
DO

```

Figure 5-7 (Part 2 of 3). Program to Find Documents with an Action Date


```

/* Send a message to the user specified in the parm */
/* of the command that a document needs action. */
CHGVAR      VAR(&MSG) VALUE(&SAVDUED *CAT +
                        ' due date for ' *CAT &SAVDNM *CAT +
                        'document in folder ' *CAT +
                        &SAVFLR)
SNDDST      TYPE(*MSG) TOUSRID((&USRID +
                        &ADDR)) DSTD('Action needed on Document') +
                        MSG(&MSG) +
                        PTY(*HIGH)

/* The record we are currently on is the record for */
/* the next document. Go to the label DUERCD to */
/* process it. */
GOTO        CMDLBL(DUERCD)
ENDDO

/* Delete the outfile created by the QRYDOCLIB */
/* command. */
EOF:        CLOF      OPNID(QAOSIQDL)
            DLTF      FILE(QTEMP/QRYDOC)

END:        RETURN    /* Return control to the calling program. */
            ENDPGM

```

Figure 5-7 (Part 3 of 3). Program to Find Documents with an Action Date

Chapter 6. Security Services

This chapter discusses how users are authorized to documents and folders. The following topics are discussed:

- How users can be authorized to documents and folders. This includes the use of authorization lists, group profiles, access codes, specific authorizations, ownership, public authorizations and special authorities. Personal documents, personal folders, and working on behalf of another user is also discussed.
- How a user's authority to a document library object is determined. A specific sequence of checks is followed. For example, if a user is specifically authorized to a document library object and the user is not working on behalf of another user, then all other ways that the user can be authorized to the object are not checked.
- Security services control language (CL) commands.
- See Chapter 2, "Calendar Services," for Calendar security information.

Security Services Overview

This section gives an overview of security services, including information on working on behalf of another user, document and folder security, and document list security.

There are two levels of system authority:

- **Object authority**

Users are authorized to objects either by ownership, by explicit authorization, or by implicit authorization through authorization lists, group profiles, access codes, public authorization, or user permission.

- **Special authority**

The user may have more authority to an object than specified for that object if the user has *special authority*. The special authorities that apply to document library object access are:

- ***ALLOBJ** All object authority is typically given to security officers. It allows the user to work with any object in the system.
- ***SECADM** Security administrator authority is typically given to OfficeVision/400 administrators. It allows the user to perform administrative-type functions such as deleting document library objects.
- ***SAVSYS** Save system authority is typically given to system operators. It allows the user to save and restore system objects.
- ***AUDIT** Audit authority is typically given to security officers or system administrators. It allows the user to set or change the level of auditing active on the system and for documents and folders.

Other ways of looking at document library object security include:

- Security as seen when using the document library object security commands
- Security as seen when using the system security commands

You should work on document library objects using document library object commands. Even though there are system security commands that allow you to work with document library objects, it is easier to work with these objects using the document library object commands. The document library object security commands are discussed in “Security Services Control Language Interfaces” on page 6-19. For example, it is better to use the DSPDLOAUT command rather than the DSPOBJAUT command to display the authorities to a library object. The DSPDLOAUT command allows you to use the user-assigned name or the system-object name of the document or folder, while the DSPOBJAUT command requires the use of the system-object name. Also, the DSPOBJAUT command does not show if this is a personal object because personal only applies to document library objects. DSPDLOAUT can be used to display any access codes associated with the document library object, where DSPOBJAUT cannot.

Note: The user-assigned name is the name given by the user to a document or folder when it is created. When the document or folder is created, the system also assigns a 10-character system-object name. See “Document Library Object Naming” on page 5-8 for more information.

Although document library object security is based on the AS/400 security model there are a number of additional concepts that have been added for document library objects. See the *Security Reference* manual for a discussion of basic AS/400 security concepts. The additional document library object concepts are:

- Working on behalf of another user (user permission).
- Access codes.
- Object authority is always one of the following types:
 - *ALL
 - *CHANGE
 - *USE
 - *EXCLUDE
- Local users must have a user profile on the system and be enrolled in the system distribution directory to access library objects.
- Remote library access requires the user to have an entry in the remote system distribution directory. Refer to “System Distribution Directory” on page 3-2 for more information.

Access Control Functions

You can manage access to documents and folders by using specific user authorization, group authorization, object ownership, public authorization, special authorities, working on behalf of others, authorization lists, and access codes. These authorization functions are supported for document library objects. Some of these functions are similar to those used for other OS/400 objects, while others are unique and supported only for document library objects. The following sections describe each of these functions and any restrictions and limitations that apply to document library object functions.

AS/400 Enrollment

A user must be enrolled in the system distribution directory before using the AS/400 services. Local users must have a user profile in order to be enrolled in the system distribution directory. Local users may use any service. However, remote users can only use the library service if they are enrolled with a user profile. Refer to Chapter 3, “Directory Services,” for more enrollment information.

CL commands used to access or administer these services require the command user to be enrolled in the system distribution directory. Certain commands do not require that the user be enrolled if the user has *ALLOBJ, *SECADM, or *SAVSYS special authority. See the *Security Reference* manual for the authority required to use a command.

User Profiles and Group Profiles

A local user must have a **user profile** before accessing the system and its resources. The profile specifies what type of access the user has to the system and how the user can use the system resources. For example, the user profile contains the *ALLOBJ, *SECADM, and *SAVSYS special authorities.

Group profiles can be created to manage sets of users. Users are associated with the group by adding the group profile name to their user profile. Once the user is associated with a group, the user has all the explicit authorities given to the group in addition to his own authorities. A user can belong to only one group at a time. The use of group profiles allows easier management of a set of users. For example, if you want to give change authority to the document DOC1 to all the users associated with the group profile GROUP1, you could use the command:

```
ADDLLOAUT DLO(DOC1) USRAUT((GROUP1 *CHANGE))
```

All the users associated with GROUP1 now have *CHANGE authority to DOC1 unless they are explicitly authorized to the document. Specific authority for a user overrides the authority of the group profile.

Specific User Authorization

Document library objects can be explicitly authorized to a specific user. This authorization includes a specific authority level for that user. Authority levels are either *ALL, *CHANGE, *USE, or *EXCLUDE. Specific object management authority or data management rights (such as *OBJMGMT or *READ rights) cannot be granted to document library objects as with other system objects.

Specific user authorization is done by using document library object commands to allow you to refer to document library objects by user-assigned names for documents and folders. System authorization commands such as Grant Object Authority (GRTOBJAUT) do not support user-assigned names. See “Security Services Control Language Interfaces” on page 6-19 for a complete list of office authorization commands.

Group Authorization

Document library objects can be explicitly authorized to a previously defined user group with the same restrictions and considerations that apply to specific user authorization. A user group is defined by enrolling a group user profile in the system distribution directory.

There are several advantages to group authorization:

- All users in the group are authorized to each object to which the group is authorized.
- Adding a user to a group implicitly authorizes that user to all objects authorized to the group. Removing a user from the group removes all authorizations.

Note: Explicit authorities are not removed.

- Group authorizations reduce the number of authorities that must be managed by the system and its users. Adding users to, or removing users from, groups is done by using system user profile management commands.

Note: Unlike other system objects, implicit ownership of a document library object *cannot* be given to a group profile, even if a user profile has the group profile specified as the owner of the objects created. Specifying the OWNER(*GRPPRF) parameter of the CRTUSRPRF or CHGUSRPRF commands does not cause document and folder ownership to be transferred to the group profile. A user owns all the documents and folders he creates, even if he is a member of a group. Also, specifying the GRPAUT parameter of the CRTUSRPRF or CHGUSRPRF commands does not set the authority for the group profile when new document library objects are created by the user.

Authorization to group profiles are also done by using unique document library commands to allow a reference by user-assigned names (see “Security Services Control Language Interfaces” on page 6-19).

Ownership Rights

Objects have only one owner. An object owner is authorized to perform any operation on the object unless rights are specifically excluded using one of the other access control functions. For example, a document owner may specifically reduce his authority level to prevent inadvertent deletion of the object. Specific exclusion does not revoke the right of the owner to change the specific authority to another level, to change the ownership of the object, or to change other authorities to the object. A document owner, for example, can always transfer ownership or change the authorizations to the document no matter what his own authority level is. Refer to “Change Document Library Object Owner (CHGDLOOWN) Command” on page 6-21 for more information on transferring the ownership of a document or folder.

Public Authorization

System objects (all documents and folders included) have a public authority. You can store the public authority either directly in the object or in an authorization list associated with the object.

Assignment of public authority to document library objects can be done when the folder or document is created or done by using unique document library commands to allow user-assigned names to refer to them. Only the CHGDLOAUT, DSPDLOAUT and EDTDLOAUT commands can be used to change or view the public authority for the root folder. (see “Security Services Control Language Interfaces” on page 6-19).

Special Authority

Special authority can be given to a user when a new user profile is created or by changing an existing user profile. The special authority allows the user to perform certain types of actions on objects that they otherwise may or may not be authorized to perform. Special authorities may be assigned to users with the SPCAUT parameter on the Create User Profile (CRTUSRPRF) or Change User Profile (CHGUSRPRF) commands. The following special authorities apply to document library objects:

- All Object (*ALLOBJ)
- Security Administrator (*SECADM)
- Save System (*SAVSYS)
- Audit (*AUDIT)

All Object Special Authority

All object (*ALLOBJ) special authority is typically given to users who perform security officer functions. Users with all object authority can access all system resources even if they have no authority to the resource. Only a user with *ALLOBJ special authority can give another user *ALLOBJ special authority. All object authority is adopted by the following commands when used in a program created by a user with all object special authority:

- Delete Document Library Object (DLTDLO)
- Save Document Library Object (SAVDLO)
- Restore Document Library Object (RSTDLO)
- Grant User Permission (GRTUSRPMN)
- Revoke User Permission (RVKUSRPMN)
- Display Authorization List Document Library Objects (DSPAUTLDLO)
- Grant Access Code Authority (GRTACCAUT)
- Revoke Access Code Authority (RVKACCAUT)
- Display Document Library Object Authority (DSPDLOAUT)
- Edit Document Library Object Authority (EDTDLOAUT)
- Change Document Library Object Authority (CHGDLOAUT)
- Add Document Library Authority (ADDDLOAUT)
- Remove Document Library Object Authority (RMVDLOAUT)
- Change Document Library Object Owner (CHGDLOWN)
- Check Document Library Object (CHKDLO)
- Reclaim Document Library Object (RCLDLO)
- Delete Document List (DLTDOCL)
- Reorganize Document Library Object (RGZDLO)

Security Administrator Special Authority

The security administrator (*SECADM) special authority allows a user to perform OfficeVision/400 administration functions for enrollment and management of the document library. The security administrator authority allows a user to:

- Add users to the system distribution directory. (This includes the right to create and change user profiles for OfficeVision/400 users.)

You must have security administrator authority to use the Create User Profile (CRTUSRPRF) and Change User Profile (CHGUSRPRF) commands. This authority is not required for the Change Profile (CHGPRF) command.

- Display authority for documents or folders.
- Change ownership of document or folders.

- Add and remove access codes to the system.
- Assign and remove a user's access code authority.
- Grant and revoke permission for users to work on another user's behalf.
- Delete documents and folders.
- Change distribution lists created by other users.
- Reclaim internal document library system objects, or reclaim all document library objects.
- Reorganize all document library objects

Only a user with *SECADM special authority can give another user *SECADM special authority through the administration option of OfficeVision/400 (STROFC). Only a user with both *ALLOBJ and *SECADM special authority can give another user *SECADM special authority when using the CRTUSRPRF or the CHGUSRPRF commands.

The security administrator special authority gives the user authority to use these commands, specific options on these commands, or use these commands to manage objects without explicit authorization to the objects:

Add Directory Entry (ADDDIRE)
 Change Directory Entry (CHGDIRE)
 Remove Directory Entry (RMVDIRE)
 Work with Directory (WRKDIR)
 Delete Distribution List (DLTDSTL)
 Delete Document List (DLTDOCL)
 Add Distribution List Entry (ADDDSTLE)
 Remove Distribution List Entry (RMVDSTLE)
 Work with Distribution List (WRKDSTL)
 Work with Document Library (WRKDOCLIB)
 Reclaim Document Library Object (RCLDLO)
 Reorganize Document Library Object (RGZDLO)
 Delete Document Library Object (DLTDLO)

Note: Security administrator special authority is not adopted by these commands when used in a program created by a user with security administrator special authority.

Save System Special Authority

Save system (*SAVSYS) special authority allows a user to save and restore documents and folders without specific object authorizations. The commands used to perform these functions are:

Save Document Library Objects (SAVDLO)
 Restore Document Library Objects (RSTDLO)

Audit Special Authority

Audit (*AUDIT) special authority gives the user the ability to change auditing characteristics. The user can:

- Change the system values that control auditing.
- Use the CHGDLOAUD command to change auditing for documents and folders.

Note: A user with this authority can stop and start auditing on the system or prevent auditing of particular actions.

Authorization Lists

Specific user and group authorizations can be stored in an **authorization list** that you can use to manage authorizations to a set of objects. For example, authorization to all objects in a folder can be managed by assigning the same authorization list to the folder and all objects in the folder. Authorizations for both office and non-office objects can be managed using the same authorization list. Only one authorization list can be specified for an object.

There are several advantages to authorization lists:

- All users in the list are authorized to an object when the list is assigned to the object.
- Adding a user to a list authorizes that user to all objects previously authorized to the list. Removing a user from the list removes the authority to objects in the list.

Note: Explicit authorities to the object are not removed.

- Authorization lists reduce the number of authorities that must be managed by both the system and its users.

Assignments of authorization lists to document library objects can be done when the object is created or by using unique document library commands to allow you to refer to documents and folders by user-assigned names (see “Access Control Functions” on page 6-2). Use authorization list management commands, such as the Edit Authorization List (EDTAUTL) command to add users to and remove users from an authorization list.

Using Authorization Lists

User and group authorizations can be stored in an authorization list which in turn can be used to manage the authorizations to a set of objects. Associating an object with an authorization list authorizes all users in the list to the object with the level specified for that user in the list. One of four authority levels is specified: *ALL, *CHANGE, *USE, and *EXCLUDE.

Adding a user profile or a group profile to the list with a specific level authorizes that user to all objects associated with the list at that level unless that user or group is also explicitly authorized to an object with a different level. That is, specific user authorizations to an object always take precedence over authorizations in the object’s authorization list.

Removing a user from the list conversely removes the list authorization.

No authority to the authorization list is required to use the list to secure an object. No authority to the list is required to identify the objects secured by the list although authorization to each object is required to view each object description. No authority to the list is required to see the users associated with the list.

Only the owner of the object, a user with *ALLOBJ special authority or a user with *ALL authority to the object can add or remove an authorization list for an object.

An authorization list can be deleted only by the authorization list owner or by a user with *ALLOBJ special authority.

Ownership of an authorization list can be transferred by the authorization list owner or by a user with *ALLOBJ special authority.

Adding, changing, and removing users associated with the authorization list requires that the requester:

- Be included in the list with a specific right called **authorization list management authority**.
- Have at least the same authority level (*ALL, *CHANGE, *USE) in the list as is associated with the entry which is to be added, removed, or changed.

Authorization list management rights are assigned to a user or group when the user or group is added to the authorization list. Authorization list management rights can be assigned only by the list owner or by another user in the list with authorization list management rights or by a user with *ALLOBJ authority.

Public Authorization and Authorization Lists

Public authorization assigns an authority level to all users not explicitly authorized to the object. Public authority levels can be *ALL, *CHANGE, *USE, *EXCLUDE, or *AUTL.

Public authority is assigned either directly to an object or indirectly to an object using an authorization list. An object secured using an authorization list can default the object public authority to the public authority of the associated authorization list. For example:

```
CHGDLOAUT DLO(xxx) USRAUT(*PUBLIC *AUTL)
```

In this case, changing the public authority in the authorization list changes the public authority of the object. An object secured by an authorization list can still have the public authority explicitly assigned in which case changing the public authority in the authorization list does not change the public authority of the object. For example:

```
CHGDLOAUT DLO(xxx) USRAUT(*PUBLIC *USE)
```

Public authority of an object is set to the current authorization list public authority level if the authorization list is detached from an object that connects public authority to an authorization list. This level is not changed if the object is subsequently attached to another authorization list.

Precedence Rules and Authorization Lists

Authority level specified in an authorization list for a given user or group is overridden by any explicit authorization level for that user or group. Authority level specified in the list for a group is overridden by an authority level specified in the list for a group member. See Figure 6-2 on page 6-17 for additional information.

Authorization List Ownership

Authorization lists are owned by the user who created the authorization list unless the user's profile specifies that a group profile is to own objects created by the user. Storage for these objects is charged against the owning user profile.

Working on Behalf of Another User

Working on behalf of another user gives you the same access to objects and services as the user on whose behalf you are working. For example, you can send and receive mail for another user, or work with all library objects that another user is authorized to access. However, you may not access distributions, documents, or folders that are marked as personal while working on another user's behalf.

When one user works on behalf of a second user, processing is done the same as if requested by the second user. Any objects created, such as documents, are owned by the second user. Access to existing documents and folders is based only on the authorizations available to the second user. That is, the second user's authorities are not added to the first user's authorities when the system performs access authorization checks. A user's own authorities are ignored when working on behalf of another user.

Permission to work on behalf of another user:

- Grants all document/folder authorization, except for objects with a personal security level
- Grants access to the mail log, except for distributions designated as personal

Removing user permission revokes these rights. See Figure 6-2 on page 6-17 for additional information.

Removing a user as an OfficeVision/400 user removes all permissions associated with the user.

Personal Items

The user sending a distribution may mark it as personal, which indicates that a user working on behalf of any recipient may not see the distribution. The owner of a filed document or folder may mark it as personal, which indicates that a user working on behalf of any other user authorized to the document or folder may not access the document or folder.

Marking an item in the document library as personal only prevents access by someone working on your behalf. It does not prevent authorized access to the library item by a user who is not working on your behalf.

For example, if a user normally has *EXCLUDE authority to a document you own that is not marked as personal, that user can access the document when working on your behalf. If you then change the document to a personal document, the user working on your behalf can no longer access that document. However, if that same user normally has *CHANGE authority to the document, then marking the document personal only prevents the user from accessing the document when that user is working on your behalf. The same user can change the document when not working on your behalf.

For documents and folders to be completely personal, you need to exclude all users from the item and also mark it personal so that users working on your behalf do not have access to the item.

Some products that attach to the AS/400 system for office services allow users working on behalf of a recipient to view a personal distribution if the recipient's document password is specified. The recipient's document password is stored in his user profile on the host AS/400 system.

Note: The document password is set up using the DOCPWD parameter of either the CRTUSRPRF or CHGUSRPRF command. The document password specified by the user working on behalf of the recipient is compared to the value stored in the recipient's user profile before access to the distribution is granted.

User Permission Commands

Permission to work on behalf of another user can be granted or revoked by using the OfficeVision/400 product or by using CL commands. The commands are:

GRTUSRPMN Grant User Permission. Allows you to grant others permission to work on your behalf. If you have *ALLOBJ authority, you can also grant permission for any user to work on behalf of any other user.

RVKUSRPMN Revoke User Permission. Allows you to remove the permission for others to work on your behalf. If you have *ALLOBJ authority, you can also remove the permission for users working on behalf of other users.

DSPUSRPMN Display User Permission. Can be used by an office user to see who can work on his behalf or on whose behalf the user is allowed to work. Can be used by an office administrator to determine who has permission to work on behalf of others:

You do not need special authority to use these commands to control which users may work on your behalf. The security officer can use these commands to grant or revoke permission for any user to work on behalf of any other user.

Permission Example

Mary, an executive, wants to give permission to Joan, a secretary, to handle her mail and receive documents. To permit Joan to work for Mary, enter the Grant User Permission (GRTUSRPMN) command. Assuming that MARY is Mary's user profile name and JOAN is Joan's user profile name, the command would be:

```
GRTUSRPMN TOUSER(JOAN) FORUSER(MARY)
```

To remove this permission, type the following:

```
RVKUSRPMN FROMUSER(JOAN) FORUSER(MARY)
```

If you remove a user from the system distribution directory, all user permissions for that user are revoked. If that user is enrolled again, the permissions must be established again.

Refer to "Security Services Control Language Interfaces" on page 6-19 for more information about user permission commands.

Refer to “Example Applications” on page 4-29 for information on granting user permission in a CL program.

Security Services for Document Library Objects

This section discusses document and folder authority levels, remotely filed documents, and the sequence of authority checking, and offers an example illustrating some of the authority checking rules.

Document and Folder Auditing

This section discusses auditing for document library objects. Refer to the *Security Reference* manual or the *Guide to Enabling C2 Security* for more information on AS/400 auditing.

The AS/400 system provides auditing support for documents and folders. This support provides the user the ability to log selective actions against specific objects and place the information in an audit journal. Actions such as object moves, ownership changes, authority changes, and object renames can be audited.

Document and Folder Security

This section discusses security for document library objects. Refer to the *Security Reference* manual for a general discussion of AS/400 security.

When you secure a document, you secure:

Document content The material that you put into the document.

Document description

Information about the document, such as descriptive text, author, and keywords. This information is assigned by the user when creating or filing a document. This information is also referred to as document details or document attributes.

When you secure a folder, you secure:

Folder table of contents

The list of items within the folder.

Folder description Information about the folder, such as descriptive text.

Unless specified when the document or folder is created:

- New documents and folders created in a folder pick up the authorities of the folder that they were created in.
- New first-level folders (folders that are not in a folder) and new documents that are not created in a folder have:
 - Public authority set to *EXCLUDE
 - No authorization list
 - Ownership set to the user who created the object
 - No other authorizations
 - Personal indicator set to NO

An object's authority is assigned at creation time and copying, moving, or removing a folder does not change its authority.

Even though a folder contains objects, a user who has authority to the folder does not necessarily have authority to the objects within the folder. Each object in the

folder has its own set of authorities that are used in determining who can access the object.

A user who has authority to a document does not necessarily have authority to the folder in which the document is contained. For example, if a user does not have change authority to the folder, the user may be able to change the contents of a document within the folder (if authorized to change the document), but not be able to add any more documents to the folder.

Creation of folders can be controlled by restricting user access to the Create Folder (CRTFLR) command. See “Create Folder (CRTFLR) Command” on page 5-23 for more information.

Creation of first-level folders can be controlled by changing the public authority on the root (*ROOT) folder. See “Edit Document Library Object Authority (EDTDLOAUT) Command” on page 6-25 and “Change Document Library Object Auditing Level (CHGDLOAUD) Command” on page 6-20 for more information.

To change the ownership of a document or folder, the user must either be the current owner of the object or have *ALLOBJ special authority.

Document and Folder Authority Levels

A user can have one of four levels of authority for each document or folder in the document library. The four levels of authority are listed as follows in the order from the most authority to the least authority:

- *ALL** The user can do all operations on the document library object except the following, which are done only by the owner or a user with *ALLOBJ authority:
- Change the ownership of the document library object
 - Give authority to work with the document library object again after that authority has been taken away
- Note:** A document library object has one owner, but many users can have *ALL authority to work with the object.
- *CHANGE** The user can change the contents and description of the document library object. For folders, the user can add objects to or move objects from the folder.
- *USE** The user can view the contents and description of the document library object, but cannot change them.
- *EXCLUDE** The user does not have authority to perform any operation on the document library object.

Figure 6-1 on page 6-14 shows the level of authority needed for various operations on documents and folders.

Figure 6-1. Operations Based on Level of Authority

| Level of Authority | Operations you can do on folders | Operations you can do on documents |
|------------------------|--|---|
| *ALL | <ul style="list-style-type: none"> • Give authority to use the folder • Take away authority to use the folder • Rename the folder¹ • Delete the folder² • Save the folder³ • Restore the folder⁴ • Do all folder operations listed for the *CHANGE and *USE levels of authority | <ul style="list-style-type: none"> • Give authority to use the document • Take away authority to use the document • Rename the document¹ • Delete the document⁵ • Save the document • Move the document⁶ • Restore the document⁴ • Do all document operations listed for the *CHANGE and *USE levels of authority |
| *CHANGE | <ul style="list-style-type: none"> • Change folder details • Add objects to the folder • Move documents from the folder⁷ • Reorganize a folder⁸ • Do all folder operations listed for the *USE level of authority | <ul style="list-style-type: none"> • Change document details • Edit the document if it is revisable⁹ • Retrieve a document and check it out • Reorganize a document⁸ • Mark the document for offline storage • Do all document operations listed for the *USE level of authority |
| *USE | <ul style="list-style-type: none"> • Read the folder details • List the directory of folders or search within a folder • Get a printout of folder storage • Display authority to a folder | <ul style="list-style-type: none"> • Read the document details • Read the document content • Copy the document¹⁰ • Retrieve a document without checking it out • Print the document • Get a printout of document storage • Display authority to the document |
| *EXCLUDE ¹¹ | <ul style="list-style-type: none"> • Reclaim a folder¹² | <ul style="list-style-type: none"> • Reclaim a document¹² |

Notes:

- 1 You also need *CHANGE authority to the folder, if any, that the folder or document is in.
- 2 To delete a folder, you must have *ALL level of authority for the folder and for each object in the folder, or *SECADM or *ALLOBJ special authority. How you specify this operation determines if the folder must be empty before deleting the specified object or if the system deletes all documents first and then deletes the folder. Objects to which you do not have authority are not deleted. The system sends a message warning you that not all objects in the folder were deleted.

You need no authority to a folder to delete an object it contains. You must have *ALL authority to the objects you want to delete, or you must have either *SECADM or *ALLOBJ special authority.
- 3 To save a folder and all objects in the folder, you must have *ALL level of authority for all objects in the folder or *SAVSYS or *ALLOBJ special authority. Objects for which you do not have authority are ignored, and the system sends a message warning you that not all objects were saved.
- 4 If the restored object replaces an existing object, you must have *ALL authority to the object or *ALLOBJ or *SAVSYS special authority. If the restored object does not exist, you must have *CHANGE authority to the folder (if any) to which the object is being restored or *ALLOBJ or *SAVSYS special authority.
- 5 You do not need authority to a folder to delete an object it contains, but you must have *ALL authority to the objects you want to delete, or have either *SECADM or *ALLOBJ special authority.
- 6 To move a document, you must have *CHANGE level of authority to both the folder (if any) from which the document is moved and the folder (if any) to which the document is moved. If the moved document does replace an existing document, you must have *ALL level of authority to the document being replaced.
- 7 To move a document, you must have *CHANGE authority to both the folder from which the document is moved and the folder to which the document is moved.
- 8 To reorganize mail or reorganize all documents and folders, you must have *ALLOBJ or *SECADM special authority.
- 9 If you are editing the document using the DisplayWrite 4 or DisplayWrite 5 programs, you must also have *CHANGE authority to the folder in which the document resides. Some PC text editors create a temporary document during editing, which replaces the original when editing is complete. Creating a document in a folder requires *CHANGE authority to the folder.
- 10 If the copied document does not replace an existing document, you must have *CHANGE level of authority to the folder (if any) to which the document is copied. If the copied document does replace an existing document, you must have *CHANGE level of authority to the document being replaced.
- 11 With *EXCLUDE authority, you have no authority to work with the folder or document.
- 12 To reclaim internal documents or reclaim all documents and folders you must have *ALLOBJ or *SECADM special authority.

Remotely Filed Documents

An AS/400 system allows documents to be filed remotely on another AS/400 system, System/38, or System/370* DISOSS. Authorization support for remotely filed documents is a function of the level of support provided by the remote system. A remotely attached AS/400 system allows all of the authorization functions described above. A remotely attached System/38 or DISOSS document library supports only authorization by access codes.

The OfficeVision/400 program supports authorization management support for remote libraries. This support includes:

- Designating a document as personal or not personal (AS/400 system and System/38 only)
- Specific user authorization (AS/400 system only)
- User group authorization (AS/400 system only)
- Authorization using a list (AS/400 system only)
- Public authorization (AS/400 system only)
- Document access code assignment

Sequence of Authority Checking

Because you can be authorized to a document or a folder in several ways and conflicts may occur, conflicts are resolved based on a sequence of **authority checking**. Authority is always checked in the same sequence. Figure 6-2 on page 6-17 shows the proper sequence of authority checking. To read the figure, start at the top and read down, taking the appropriate branch at each decision point. The first time a branch leads to a specific authority level (*ALL, *CHANGE, *USE, or *EXCLUDE), the checking stops.

Note: When document and folder objects are checked for authority, the AS/400 system recognizes special authorities only for programs that adopt the owner's authority. All other authority checking is done for the user of the program.

Authority Checking Example

The following example illustrates some of the authority checking rules.

John Smith is the manager of two purchasing groups that share access to large numbers of documents and folders. To facilitate this sharing, the following is created:

Group user profile PURGRP1 with members TOM, PEGGY, and SUE

Group user profile PURGRP2 with members MARY, TIM, JIM, and CONNIE

Authorization list PURCHASE containing:

| User or Group | Authorization |
|---------------|---------------|
| PURGRP1 | *USE |
| PURGRP2 | *USE |
| JOHN (mgr) | *USE |

Users PEGGY and JIM want to create a new policy proposal for purchasing. While they are changing and improving it, they want members of both purchasing groups to review and critique it. However, they do not want John, the manager, to see it until they have it completed. Peggy creates a new document and assigns the following authorization to it:

Authorization list = PURCHASE

Peggy also specifically excludes John.

| User or Group | Explicit Authorization |
|-----------------|---------------------------|
| ----- | ----- |
| JIM (co-author) | *ALL |
| JOHN (mgr) | *EXCLUDE |

When the proposal is complete, the *EXCLUDE entry for JOHN can be removed.

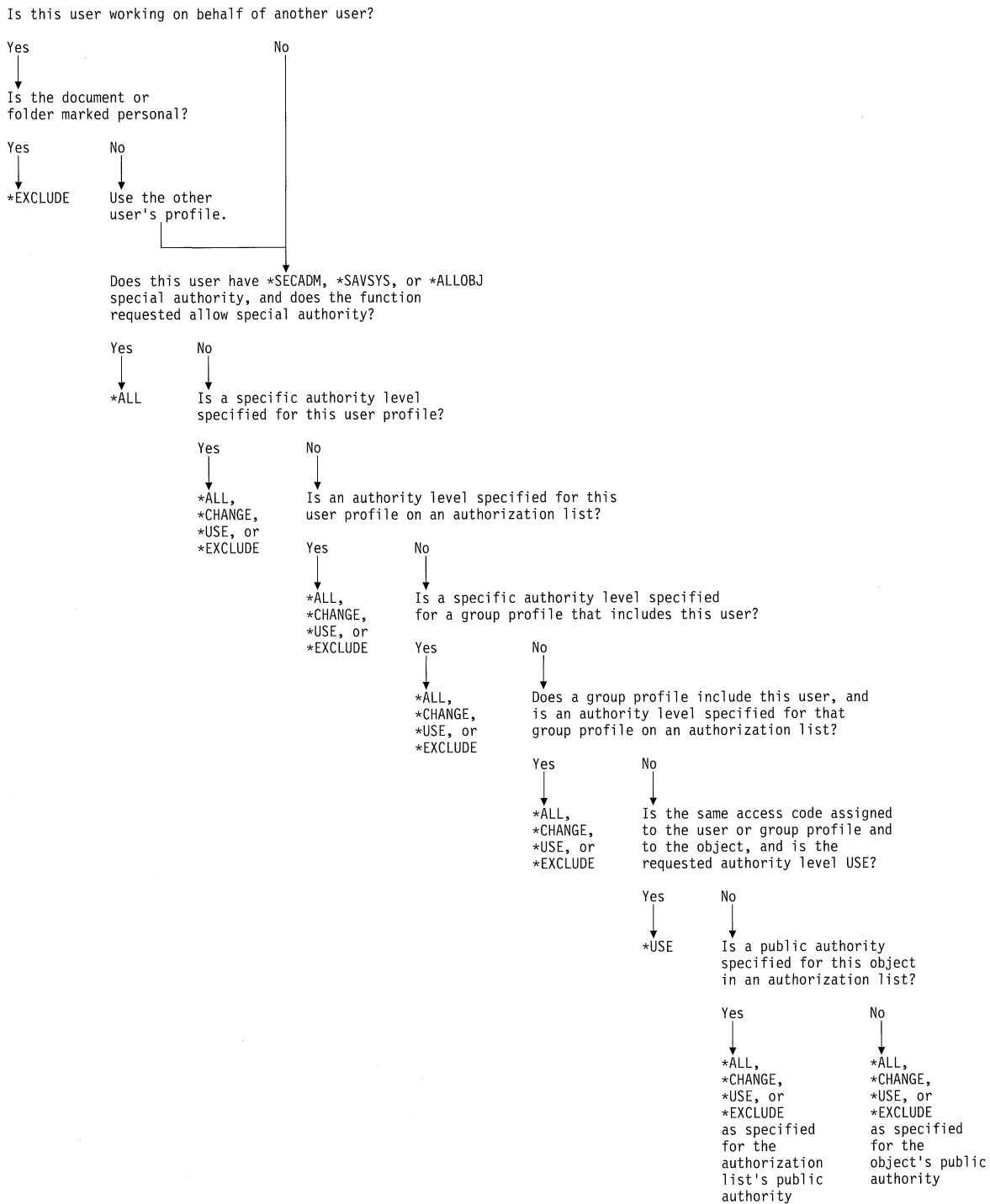


Figure 6-2. Authority Checking Chart

Access Codes

Access codes are the only method to secure documents stored remotely on System/38 or System/370 DISOSS and are supported by the AS/400 system for coexistence with these systems. If coexistence with these systems is not required, other access control methods, such as authorization lists, are recommended.

You can authorize a group of users to document library objects by associating a predefined access code with the object and with all users allowed to access the object. Access code authorization is unique to document library objects and is not supported for other system objects.

Users authorized to a document or folder through a 4-digit **access code** are only authorized with *USE authority. Access codes can be associated with a specific user or group and multiple access codes can be assigned to an object. A maximum of 2047 access codes can be defined on an AS/400 system. A user can access an object protected by access codes if the user or user's group is assigned to any access code that secures the object.

Access code authorization is done using the following office security commands:

| | |
|------------------|--|
| ADDACC | Add Access Code adds a new access code to the system. |
| DSPACC | Display Access Code displays all existing access code on the system. |
| DSPACCAUT | Display Access Code Authority displays access codes associated with a user. |
| GRTACCAUT | Grant Access Code Authority grants *USE authority for a user to all document library objects that are associated with the specified access code. |
| RMVACC | Remove Access Code removes an existing access code from the system. |
| RVKACCAUT | Revoke Access Code Authority revokes authority for a user to all document library objects associated with the specified access code. |

See the *CL Reference* manual for the format of these commands.

Security Services Control Language Interfaces

Use the commands described in this section to access security services functions. For more information on these and other CL commands, see the *CL Reference* manual. The following commands are described:

- Add Document Library Object Authority (ADDDLOAUT)
- Change Document Library Object Auditing Level (CHGDLOAUD)
- Change Document Library Object Authority (CHGDLOAUT)
- Change Document Library Object Owner (CHGDLOOWN)
- Display Authorization List Document Library Objects (DSPAUTLDLO)
- Display Document Library Object Auditing Level (DSPDLOAUD)
- Display Document Library Object Authority (DSPDLOAUT)
- Display User Permission (DSPUSRPMN)
- Edit Document Library Object Authority (EDTDLOAUT)
- Grant User Permission (GRTUSRPMN)
- Remove Document Library Object Authority (RMVDLOAUT)
- Revoke User Permission (RVKUSRPMN)

Add Document Library Object Authority (ADDDLOAUT) Command

The Add Document Library Object Authority (ADDDLOAUT) command gives users access to document library objects. You can grant authority either to a specific user or to a group of users through use of access codes or authorization lists. This command gives one or more users authorization to a document or folder in the following ways:

- By associating an authorization list with the document or folder
- By explicitly authorizing up to 50 user profiles or group profiles to the document or folder
- By associating up to 50 access codes with the document or folder

Security Considerations

You must have *ALL authority to the objects, *ALLOBJ special authority, or be the owner of the object to use this command.

Some Ways to Use This Command

Use the ADDDLOAUT command to:

- Authorize a user to one or more document library objects
- Authorize a user to all the objects in a folder
- Authorize all the users associated with a group profile to all the objects in a folder
- Authorize a set of users associated with an authorization list to a folder or document

Note: Only one authorization list can be associated with a document or folder.

You may use either a user-defined name or a system-object name with this command.

A group of users can be authorized to all the objects in a folder. All the users have the same authority to the objects in the folder. This does not set the authority of the group to the folder itself.

```
ADDLOAUT DLO(*ALL) FLR(folder-name) USRAUT((group-profile-name authority))
```

Change Document Library Object Auditing Level (CHGDLOAUD) Command

The Change Document Library Object Auditing Level (CHGDLOAUD) command allows a user with *AUDIT special authority to change the auditing level of a document library object (DLO) or a group of DLOs.

This command changes an existing auditing level of a document library object. Only the document library objects that exist when the command begins processing are changed. A user can change the auditing levels for the following:

- The objects in a folder
- The document or folder with a system object name
- The folder
- The root (*ROOT) level folder

Note: The root (*ROOT) level folder is changed first when changing all objects on the system.

Security Considerations

You must have *AUDIT special authority to use this command.

Some Ways to Use This Command

Use the CHGDLOAUD command to:

- Change the auditing level of a document in a folder.
- Change the auditing level of a document with a system object name so all change or read access to the DLO is logged in the security journal.

The following example changes the auditing level of all DLOs. All change or read access to the DLOs is logged in the security journal.

```
CHGDLOAUD DLO(*ALL) FLR(*ANY) DLOAUD(*ALL)
```

Change Document Library Object Authority (CHGDLOAUT) Command

The Change Document Library Object Authority (CHGDLOAUT) command changes an existing user's authority to a document library object. You can do this by changing the specific authority a user has to an object. You can also do this by changing public authority or by changing the authorization list that specifies an object's security.

This command changes an existing authorization to a document, folder, or the public authority for the root folder. You can change the following:

- Authorization list associated with the document or folder
- Up to 50 explicit user-profile or group-profile authorizations to the document or folder
- Personal status of the document or folder
- Public authority for the document or folder

Note: Personal status and public authority cannot be added to or deleted from a document, folder, or root folder.

Security Considerations

You must have *ALL authority to the object, *ALLOBJ special authority, or be the owner of the object to use this command.

When USRAUT((*PUBLIC *CHANGE)) and DLO(*ROOT) are specified, all users can create first-level folders in the root folder. When USRAUT((PUBLIC *USE)) with DLO(*ROOT) is specified, only users with *ALLOBJ or *SECADM authority can create first-level folders. Folder creation is the only function controlled by these values. Public authority is the only security value that can be specified for the root (*ROOT) folder. Only *CHANGE and *USE public authorities can be specified for the root (*ROOT) folder.

Some Ways to Use This Command

Use the CHGDLOAUT command to:

- Change a user's authority to all the objects in a folder
- Change a group of user's authorities to a document by changing the group profile's authority
- Authorize a different set of users to a document or folder by changing the authorization list

Note: Only one authorization list can be associated with a document or folder.

You can use a user-defined name or a system-object name with this command.

The REFDLO parameter can be used to replace all existing authorities to the document or folder with the authorities from another document or folder. For example, if you create a new document named NEWDOC and you have another document named OLDDOC, you can authorize all the users authorized to OLDDOC to also be authorized to NEWDOC by using the following command:

```
CHGDLOAUT DLO(NEWDOC) REFDLO(OLDDOC)
```

The following example gives all users authority to create first-level folders in the root (*ROOT) folder.

```
CHGDLOAUT DLO(*ROOT) USRAUT((*PUBLIC *CHANGE))
```

Change Document Library Object Owner (CHGDLOOWN) Command

The Change Document Library Object Owner (CHGDLOOWN) command transfers object ownership from one user to another user. The second user becomes the new owner. The authorities that other users have to the document do not change. The previous owner's authority is revoked, unless otherwise specified.

Documents saved with STG(*FREE) can have the ownership changed.

The owner of the object has *ALL authority for the object unless it is specifically changed. The owner has the authority to change any user's authority for the object and can grant himself any rights that were specifically revoked. For example, the owner can reduce his authority as a precaution. When he needs the authority, the owner grants himself those same rights again.

The user to whom ownership is being transferred must be in the system distribution directory.

Note: If the user profile specifies ownership, the system does not check a group profile to determine ownership for a document or folder.

Security Considerations

To transfer ownership of a document or folder, you must have *ALLOBJ special authority, or be the owner of the document or folder.

Some Ways to Use This Command

The previous owner's authority to the documents or folders is revoked unless the CUROWNAUT(*SAME) parameter is used.

All the objects that belong to a user can be transferred to another user by using the OWNER parameter of this command. For example, to transfer all the documents and folders from user JANE to user MARY, enter the following command:

```
CHGDLOOWN OWNER(JANE) NEWOWN(MARY)
```

Note: This revokes Jane's current authorization to the objects. To keep Jane's authority use:

```
CHGDLOOWN OWNER(JANE) NEWOWN(MARY) CUROWNAUT(*SAME)
```

Display Authorization List Document Library Objects (DSPAUTLDLO) Command

The Display Authorization List Document Library Objects (DSPAUTLDLO) command lists the documents and folders whose security is specified by the authorization list in the AUTL parameter.

Documents and folders are listed with the user-assigned names. The output of this command can either be displayed or sent to a spooled file for printing.

Note: Other system objects can be associated with this list. Use the DSPAUTLOBJ command to display these objects.

Security Considerations

If you are on the list with authority other than *EXCLUDE or not on the list and the public authority is other than *EXCLUDE, you are authorized to display the documents and folders associated with the list.

If you are not authorized to a document or folder in the list, the document or folder name is displayed and description of the document or folder will be *NOT AUTHORIZED.

Some Ways to Use This Command

The output of this command can be either displayed or sent to a spooled file to be printed.

To print the documents and folders associated with an authorization list enter:

```
DSPAUTLDLO AUTL(aut1-name) OUTPUT(*PRINT)
```


Display Document Library Object Auditing Level (DSPDLOAUD) Command

The Display Document Library Object Auditing Level (DSPDLOAUD) command allows a user to display the auditing level of a document or a folder. The audit information can be shown on the display, printed with the job's spooled output, or directed to a database file. See Figure C-5 on page C-39 for the layout of the output file.

This command displays an existing auditing level of a document library object. A user can display the auditing levels for the following:

- The objects in a folder
- The document or folder with a system object name
- The folder
- The root (*ROOT) level folder

Security Considerations

You must have *USE authority to the document or folder or have *ALLOBJ or *AUDIT special authority to display the auditing level of the document or folder.

You must be enrolled in the system distribution directory unless you have *ALLOBJ or *AUDIT special authority.

Some Ways to Use This Command

Use the DSPDLOAUD command to:

- Display the auditing level of a document in a folder
- Display the auditing levels of all DLOs on the system in an output file

The following example prints the auditing levels of all first-level folders on the system.

```
DSPDLOAUD DLO(*ALL) FLR(*ROOT) OUTPUT(*PRINT)
```

Display Document Library Object Authority (DSPDLOAUT) Command

The Display Document Library Object Authority (DSPDLOAUT) command lists the authorized users of an object and their assigned rights of use. The list includes the following information:

- Name of the document library object and folder (if any)
- Owner of the document library object
- Name of the authorization list securing the document or folder
- Personal status of the document or folder
- Checkout status of the document (does not apply to folders)
- Specific user authority for the document or folder
- The authority given to the users with no specific authority for the document or folder
- Access codes associated with the document or folder

The access codes attached to the document library object can be displayed with a function key.

The document library objects and users associated with an authorization list, if any, used to secure the document or folder can be displayed with a function key.

Security Considerations

A user must have *USE authority to the document or folder to display that user's authority information about the document or folder.

If a user has *ALL or *ALLOBJ authority to the document or folder, then all users and their authorities appear. If you have authority other than *EXCLUDE, only your authority appears. If you have *EXCLUDE authority, the following message appears:

No authority to view or change the security of document library object ____.

A user must have *ALLOBJ or special authority to display the root (*ROOT) folder public authority.

Some Ways to Use This Command

Either a user-defined name or a system-object name can be used with this command.

The output of this command can be either displayed or sent to a spooled file to be printed.

To print the authorizations for a document or folder, enter:

```
DSPDLOAUT DLO(DLO-name) FLR(folder-name) OUTPUT(*PRINT)
```

The owner, authorization list, and personal values do not appear if DLO(*ROOT) is specified. The function keys that allow a user to view authorization list contents, access codes, and explicit user authorizations are disabled. To display the public authority for the root (*ROOT) object, enter:

```
DSPDLOAUT DLO(*ROOT) FLR(*NONE) OUTPUT(*)
```

Display User Permission (DSPUSRPMN) Command

The Display User Permission (DSPUSRPMN) command displays the users who:

- Have permission to handle documents or folders and perform office-related tasks on behalf of another user
- Have permitted other users to work on their behalf

You may also use this command to display a list of all users and their permissions for up to 300 specific users.

Security Considerations

Any user can use this command.

Some Ways to Use This Command

Use the DSPUSRPMN command to:

- List users who can work on behalf of a user
- List users on whose behalf the user can work
- List all users and the users who can work on their behalf (requires *SECADM or *ALLOBJ special authority)
- List all users and the users on whose behalf they can work

The output of this command can either be displayed or sent to a spooled file to be printed.

To see all the users with permission to work on behalf of another user, enter:

```
DSPUSRPMN USER(*ALL) GRANTED(*FROM)
```

To see all the users who have granted other users permission to work on their behalf, enter:

```
DSPUSRPMN USER(*ALL) GRANTED(*TO)
```

Edit Document Library Object Authority (EDTDLOAUT) Command

The Edit Document Library Object Authority (EDTDLOAUT) command changes authorization to a document or folder.

The following information is displayed for the specified document or folder:

- Name of the document library object and folder (if any) in which the document is filed
- Owner of the document library object
- Name of the authorization list securing the document or folder
- Personal status of the document library object
- Checkout status of the document (does not apply to folders)
- Specific user authority for the document or folder
- Authority given to the users who have no specific authority for the document or folder
- Access code associated with the document or folder

The access codes attached to the document library object can be displayed with a function key.

The document library objects and users associated with the authorization list, if any, used to secure the document library object can be displayed with a function key.

Security Considerations

You must have *ALL authority to the document or folder, *ALLOBJ special authority, or be the owner of the document or folder to change the object's authority with this command.

| Only a user with *ALLOBJ special authority can modify the public authority value for
| the root (*ROOT) folder. The values *CHANGE and *USE are the only ones that
| can be specified for the public authority parameter. If the root folder public
| authority value is set to *CHANGE, all users can create first-level folders. This is
| the default value. If the root folder public authority value is set to *USE, only users
| with *ALLOBJ or *SECADM special authority can create first-level folders. Other
| users receive authority error messages when they attempt to create a first-level
| folder.

Some Ways to Use This Command

The EDTDLOAUT command can only be used interactively. Use ADDDLOAUT or CHGDLOAUT for batch. Either a user-defined name or a system-object name can be used with this command.

Grant User Permission (GRTUSRPMN) Command

The Grant User Permission (GRTUSRPMN) command gives a user permission to handle documents and folders or to perform office-related tasks on behalf of another user. The specified user is allowed access to documents, folders, and mail that are not personal. For example, if user A gives user B permission to work on behalf of user A, user B can obtain, file, delete, retrieve, list, distribute, cancel, and search a document library on behalf of user A. However, user B cannot get user A's personal mail, personal documents, and personal folders from the library. Personal mail, documents, or folders can be read or used only by the recipient or the owner of the document or folder. **Personal mail** can be accessed only by the receiver, not by someone working on the behalf of the user. A **personal document** can be read or used only by its owner, and a **personal folder** can be used only by its owner.

Both users must be in the system distribution directory. Use the Work with Directory (WRKDIR) command to enroll users.

Security Considerations

A user with *ALLOBJ special authority can grant permission for a user to work on behalf of someone else. Otherwise, a user may only grant permission on his own behalf. Granting permission to a group profile does not grant permission to all the users associated with that group.

Some Ways to Use This Command

The user specified in the TOUSER field can work on behalf of up to 300 users at a time.

Remove Document Library Object Authority (RMVDLOAUT) Command

The Remove Document Library Object Authority (RMVDLOAUT) command removes an existing user's authority to documents or folders.

You can remove the following types of authority:

- Specific user's authority
- Authorization list's authority to the associated document library object
- Object access code associated with the document library object

Security Considerations

The user of this command must have *ALL authority to the objects, be the owner of the objects, or have *ALLOBJ special authority.

Some Ways to Use This Command

Use the RMVDLOAUT command to:

- Delete a user's authority to all objects in a folder
- Delete a set of user's authorizations that come from an authorization list by removing the authorization list associated with the document or folder
- Delete a set of users authorizations associated with a group profile by removing the group profile's authority

You may use either a user-assigned name or a system-object name with this command.

A group of users can be unauthorized to all the objects in a folder. This does not remove the group's authority to the folder itself.

```
RMVDLOAUT DLO(*ALL) FLR(folder-name) USER((group-profile-name))
```

Revoke User Permission (RVKUSRPMN) Command

The Revoke User Permission (RVKUSRPMN) command removes permission from one user (or all users) to access documents and folders on behalf of another user.

If you call this command at the time work is being done on behalf of another user, any functions started are completed. However, additional functions are not accepted.

Security Considerations

A user with *ALLOBJ authority can revoke the permission of a user to work on behalf of another user.

Some Ways to Use This Command

If work is being done on behalf of another user at the time RVKUSRPMN is run, any functions currently being run are completed. Additional functions are not accepted.

Example Applications

The following is an example of how you can use the CHKDLO command to determine the authority of a user to the documents in a folder.

```
          PGM          PARM(&DEN &DGN &CHKFLR)
/*=====*/
/* This program displays a list of documents in a folder and */
/* lists the user's authority to each document. For example, */
/* if folder ABC contains three documents and a valid user is */
/* specified then the program will display the following:    */
/*   User has *EXCLUDE authority to DOC2                    */
/*   User has *CHANGE  authority to DOC1                    */
/*   User has *ALL     authority to DOC3                    */
/* The caller of this program must have *ALLOBJ special    */
/* authority.                                              */
/*                                                        */
/* The program parameters are:                            */
/*   &DEN   - user id of the user that you wish to display */
/*           authority for                                  */
/*   &DGN   - address of the user that you wish to display */
/*           authority for                                  */
/*   &CHKFLR - the folder that contains the documents that you */
/*           want a list of. You must enter this folder    */
/*           name in quotes with 63 characters specified.  */
/*                                                        */
/* Example call to this program:                            */
/*   CALL CHKFLR PARM(CHARLIE CHICAGO 'OFCFLR              */
/*                   ')                                     */
/* CHARLIE is the user or destination element name (DEN)  */
/* and CHICAGO is the address or destination group name   */
/* (DGN) and OFCFLR plus 57 blank characters is the       */
/* folder name. See CL Programmer's Guide for rules for   */
/* passing parameters with a length of greater than 32.   */
/*                                                        */
/* It is recommended to call this program from a command  */
/* (like the example command CHKDOCAUT) because the       */
/* folder name needs to be in a 63 character quoted string. */
/*                                                        */
/*=====*/
```

Figure 6-3 (Part 1 of 3). Program Using the CHKDLO Command to Determine Authority

```

/*****
/* Declare program parameters.
/*****
      DCL      VAR(&DEN) TYPE(*CHAR) LEN(8)
      DCL      VAR(&DGN) TYPE(*CHAR) LEN(8)
      DCL      VAR(&CHKFLR) TYPE(*CHAR) LEN(63)
/*****
/* Set up variables needed by this program.
/*****
      DCL      VAR(&DSPVAL) TYPE(*CHAR) LEN(8)
      DCL      VAR(&MSGTEXT) TYPE(*CHAR) LEN(60) +
              VALUE('User has          authority to +
                    document')
      DCL      VAR(&DOCNAME) TYPE(*CHAR) LEN(8)
      DCL      VAR(&AUTNUM) TYPE(*DEC) LEN(1 0)
/*****
/* Declare outfile needed by DSPFLR command.
/*****
      DCLF      FILE(QSYS/QADSPDOC) RCDFMT(DOCDTL)
/*****
/* Get a list of documents in the folder.
/*****
      DSPFLR     FLR(&CHKFLR) TYPE(*DOC) +
              OUTPUT(*OUTFILE) OUTFILE(QTEMP/OUTCHKFLR)
      MONMSG     MSGID(OFC8008) EXEC(GOTO CMDLBL(FLRERROR))
/*****
/* Now override the system file with the file just created by
/* DSPFLR. This lets us easily access the record format of
/* the file.
/*****
      OVRDBF     FILE(QADSPDOC) TOFILE(QTEMP/OUTCHKFLR)
/*****
/* Open outfile.
/*****
      OPNDBF     FILE(QSYS/QADSPDOC) OPTION(*INP)
/*****
/* Get each document from the file and determine the user's
/* authority to the document. Display the user's authority.
/*****
      /* Read the records sequentially from the DSPDIR
      /* outfile.
READF:         RCVF      RCDFMT(DOCDTL)

      /* Monitor for EOF message
      MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

```

Figure 6-3 (Part 2 of 3). Program Using the CHKDLO Command to Determine Authority

```

/* Initialize variables needed to process each */
/* document in the DSPFLR output file. */
CHGVAR VAR(&AUTNUM) VALUE(0)
CHGVAR VAR(&DSPVAL) VALUE('*EXCLUDE')

CHKAUT: /* Check for specified authority to the document */
CHKDLO DLO(&DOCNAM) FLR(&CHKFLR) +
AUT(&DSPVAL) USRID(&DEN &DGN)

/* Program will end if CHKDLO command failed message */
/* is received. */
MONMSG MSGID(CPF8A11) EXEC(GOTO CMDLBL(FAIL))

/* If the user's authority is not the specified */
/* authority in &DSPVAL, check the other authorities. */
MONMSG MSGID(CPF8A83) EXEC(GOTO CMDLBL(CHKOTHER))
/* Otherwise, the user has the specified authority */
/* to the document. */
GOTO SHOWIT

/* Control comes to here from the MONMSG of the */
/* CHKDLO command if the user did not have the */
/* specified authority to the document. */
CHKOTHER: CHGVAR VAR(&AUTNUM) VALUE(&AUTNUM+1)
IF (&AUTNUM *GT 3) (GOTO CMDLBL(ERROR))
IF (&AUTNUM *EQ 1) (CHGVAR VAR(&DSPVAL) VALUE('*ALL '))
IF (&AUTNUM *EQ 2) (CHGVAR VAR(&DSPVAL) VALUE('*CHANGE '))
IF (&AUTNUM *EQ 3) (CHGVAR VAR(&DSPVAL) VALUE('*USE '))
GOTO CHKAUT

/* Display message with user's authority. */
SHOWIT: CHGVAR VAR(%SST(&MSGTEXT 10 8)) VALUE(&DSPVAL)
CHGVAR VAR(%SST(&MSGTEXT 41 12)) VALUE(&DOCNAM)
SNDPGMMSG MSG(&MSGTEXT)

/* Read another record from the DSPFLR outfile. */
NEXTDOC: GOTO CMDLBL(READF)
/*****
/* Error reporting code. */
*****/
FAIL: SNDPGMMSG MSG('CHKDLO failed')
GOTO CMDLBL(EOF)

FLRERROR: SNDPGMMSG MSG('Not authorized to folder ' *CAT &CHKFLR)
GOTO CMDLBL(EXIT)

ERROR: SNDPGMMSG MSG('Programming error - Authority to the +
document should be *EXCLUDE, *ALL, *CHANGE +
or *USE')

EOF: CLOF OPNID(QADSPDOC)
EXIT: ENDPGM

```

Figure 6-3 (Part 3 of 3). Program Using the CHKDLO Command to Determine Authority

The following is an example command that calls the previous example program:

```

/*****/
/*
/* COMMAND NAME: CHKDOCAUT
/*
/* COMMAND TITLE: Check Document Authority
/*
/* DESCRIPTION: Check document authority within a folder
/*
/* To create this command, use the CRTCMD command. This is the
/* source file for the command.
/*
/*****/
CHKDOCAUT: CMD  PROMPT('Check Document Authority')
               PARM  KWD(USRID)  +
                  TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) MAX(1) +
                  EXPR(*YES) FILE(*NO) VARY(*NO) PASSATR(*NO) +
                  PROMPT('User ID')
               PARM  KWD(ADDR)  +
                  TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) MAX(1) +
                  EXPR(*YES) FILE(*NO) VARY(*NO) PASSATR(*NO) +
                  PROMPT('User Address')
               PARM  KWD(FLR)  +
                  TYPE(*CHAR) LEN(63) RSTD(*NO) MIN(1) MAX(1) +
                  EXPR(*YES) FILE(*NO) VARY(*NO) PASSATR(*NO) +
                  PROMPT('Folder name')

```

Figure 6-4. Check Document Authority Command

Chapter 7. Text Search Services

This chapter describes text search services, discusses objects, describes linguistic processing and how to change stop word lists, lists the text search services control language (CL) interfaces, and provides programming examples.

Text Search Services Overview

This section describes text search services. The overview includes information about:

- Text search
- Text index
- Scheduling queue
- Administration
- Language dictionaries
- Linguistic processing
- Search functions

Text Search

Text search is an optional document search tool that can be included in the installation of the OfficeVision/400 program.

Text search complements the document search function available with OfficeVision/400. The document search function provided with OfficeVision/400 locates documents according to their details, such as subject, author, or date. Text search locates documents that have a specific phrase, word, word part, or a combination of words or phrases in the text of the document.

Document searches can be based on document details, document content, or both. A typical example would be a search that locates only those documents that contain specific text written by a specific author.

Text Index

The **text index** is built from words and phrases of the documents that are chosen from the OS/400 library. Text search uses the text index in searching for documents when you enter phrases, words, or word parts. Text search builds the initial text index and updates the text index as information about documents is added, deleted, or changed. The process of extracting words and word parts from the text of a document and adding this information to the text index is called **indexing**. For more information about the text index, see "Text Index" on page 7-3.

Scheduling Queue

The **scheduling queue** contains the names of documents to be indexed. The documents, selected from the document library, are those that users have requested be included in, updated in, or deleted from the text index.

Requests can be submitted to the scheduling queue using the following CL commands:

- Add Text Index Entry (ADDTXTIDXE)
- Remove Text Index Entry (RMVTEXTIDXE)

For more information about the scheduling queue, see “Scheduling Queue” on page 7-4.

Administration

Administration functions allow you to maintain the text index, use the **administration monitor** to run automatic administration of the text index, and display and change administration details. For more information about the administration monitor, see “Automatic Administration” on page 7-5.

To maintain the text index, you use functions to update the index and reorganize the index. Updating the index processes all requests on the scheduling queue. These requests change the text index by either adding, updating, or removing the scheduled document text information. Reorganizing the index merges and condenses text index information that has been added during earlier index updates.

You can perform administration functions using the following CL commands:

- Start Index Monitor (STRIDXMON)
- End Index Monitor (ENDIDXMON)
- Display Text Index Status (DSPIDXSTS)
- Start Update of Index (STRUPDIDX)
- Start Reorganization of Index (STRRGZIDX)
- Work with Text Index (WRKTEXTIDX)

For information about how to administer text search, see the *Managing OfficeVision/400** manual.

Language Dictionaries

Text search indexes documents that are written in a national language such as English, French, or German. The document indexing procedure analyzes the text content of the selected document and stores selected words in the text index. To analyze the document text, text search uses the language dictionary indicated by the document’s language identifier if the dictionary is available from Licensed Program Language Dictionaries/400 (5738-DCT). The **language identifier** (language ID) is a 3-character identifier that is stored in a document.

For information about language dictionaries in OfficeVision/400, see “Dictionaries” on page 8-2. For information about dictionaries in text search, see “Language Dictionaries” on page 7-7.

Linguistic Processing

Linguistic processing divides text into words and sentences, selects the keywords from the text, and processes the keywords to get the base forms for keywords, their synonyms, and compound word components. Linguistic processing uses OfficeVision/400 dictionaries and stop word lists for each language to store word information, find keywords, and return the keywords in a standard, consistent form. For details about linguistic processing, see “Linguistic Processing” on page 7-8.

Search Functions

The document **search function** provided by the OfficeVision/400 program and the Query Document Library (QRYDOCLIB) CL command allows you to locate documents. You provide one or more phrases, words, or word parts and do the following in a document search:

- Match the phrase word order exactly or phrase words in the same sentence
- Optionally use synonyms to match words with similar meanings

Search functions can be run while the text index is being updated or reorganized. This allows full use of the search functions while new document information is being added to the text index. For information about searching for documents, see the *Using OfficeVision/400** manual.

Text Search Services Objects

This section describes the text search services objects, including the text index, scheduling queue, administration monitor, and files created by text search.

Text Index

The text index consists of the following internal parts:

- A **primary index** that contains information for all documents indexed before the last reorganization of the text index.
- An **additional index** that contains information for documents indexed since the last reorganization.

The text index is created and maintained as follows:

- The first index update run after installing text search creates the primary index. After the index is created, documents that have been changed or added to the library but have not yet been indexed are not in the index.
- Later index updates create or add to the additional index.

When an update takes place, information about new or changed documents for which indexing is requested is added to the additional index. References to deleted documents are marked as not available. When the update is finished, the updated additional index is included in the search process and the new index information is made available for searches.

The process of updating the additional index takes longer as the index grows. This reduces the efficiency of both the text search functions and the index update process.

- Text index reorganizations merge the primary and additional indexes, leaving only a primary index, and reset the size of the additional index to zero.

Reorganize the additional index with the primary index periodically to form a new primary index of all documents in the library. Reorganization restores the maximum efficiency of searches and index updates.

Text searches are allowed during the reorganization and use all index information.

- Index updates and reorganizations occur as scheduled or requested to maintain the text index.

There are two ways to update or reorganize the text index:

- Use an administration monitor to request updates and reorganization of the text index automatically at preset intervals. For information about the administration monitor, see “Automatic Administration” on page 7-5.
- Use the following CL commands to request manual index updates or reorganizations at any time:
 - Start Update of Index (STRUPDIDX)
 - Start Reorganization of Index (STRRGZIDX)

Index updates and index reorganizations run as jobs in QBATCH.

Scheduling Queue

The **scheduling queue** is a list of requests for documents to be included in, updated in, or deleted from the text index. Index updates process all requests on the scheduling queue. Processing these requests changes the text index by adding, updating, or removing the scheduled document text information. A document search request does not find the current contents of documents on the scheduling queue until an index update has processed these requests.

When users add or change documents in the OS/400 library, they can request that the documents be indexed. When users delete indexed documents from the OS/400 document library, the index entries must be deleted from the text index. These index update requests are placed in the scheduling queue.

Only the most recent request to index a document processes during text index updates. This ensures that documents are only processed once for each text index update.

If an index update involves a very large amount of text, it is divided into several smaller, intermediate updates. Each intermediate update uses a subset of the documents on the scheduling queue. The contents of those documents are available for searches when the intermediate update is complete. Also, an automatic reorganization is done after each intermediate update to improve the performance of the later updates.

The indexing process uses the Document Content Architecture form of the document (RFTDCA or FFTDCA type). If the Document Content Architecture form exists when indexing of the document begins, it is used. If it does not exist and there is a document conversion for the document type to convert it into the RFTDCA or FFTDCA form, the conversion is done. For RFTAS400 and FFTAS400 document types, the document content is converted to the RFTDCA or FFTDCA form. Both forms exist while the entire scheduling queue is being processed, so more storage is needed for all documents being indexed until the indexing process is complete. For converted documents, the storage used can increase up to 50 percent until the processing of all requests on the queue is complete. At the end of the indexing process, the Document Content Architecture form is deleted if it was

created for use by text search. Plan for this temporary storage if documents on your system will be converted.

You may need to clear the objects in the scheduling queue during a general recovery action following a serious error, or you may have erroneously selected documents and want to delete them from the queue.

Automatic Administration

Text search uses an **administration monitor** that automatically updates and reorganizes the index at preset intervals. The administration monitor uses a **polling frequency**. This is the frequency at which preset conditions that determine automatic administration are checked.

To determine when to update the index, the administration monitor checks the following conditions at the polling frequency that you set:

- The amount of time to wait before determining if an update is required
- The number of documents in the scheduling queue

When one of the conditions reaches its preset value, the administration monitor requests an update of the text index.

To determine when to reorganize the text index, the administration monitor checks the following conditions at the polling frequency that you set:

- The amount of time to wait before determining if a reorganization is required
- The number of documents in the additional index

When *both* of the conditions reach a preset value, the administration monitor requests a reorganization of the text index.

The administration monitor runs in the background on the batch machine, QBATCH. The administration monitor has to wait its turn based on QBATCH restrictions, regardless of the preset polling frequency intervals.

Text Search Services Files

When you install text search, library QBBCSRCH contains all text search programs and other objects. Text search creates and uses working files while performing functions such as searching, updating the text index, and reorganizing the text index. In each file, text search creates member names according to an internal algorithm. The following working files are stored in library QBBCSRCH:

| | | | | |
|-----------|----------|----------|----------|----------|
| EHWALN1 | EHWSW06 | EHWSDQ01 | EHWERRRD | EHWWOR5 |
| EHWSDD01 | EHWBLSRT | EHWURECT | EHWSDW01 | EHWMESWR |
| EHWSW04 | EHWSDJ01 | EHWDLO1 | EHWWOR3 | EHWSWO2 |
| EHWAL01 | EHWSWO7 | EHWSDR01 | EHWEXIN | EHWWOR6 |
| EHWSDELTA | EHWDELTA | EHWWOR1 | EHWSRECT | EHWMMSG |
| EHWSW05 | EHWSDLO | EHWERR | EHWWOR4 | EHWSWO3 |
| EHWBLSRO | EHWUNW | EHWSDS01 | EHWIAMSG | EHWWOR7 |
| EHWSDI01 | EHWDLN1 | EHWWOR2 | EHWWOR1 | EHVLRVS |
| EHWRDOUT | EHWRDTAG | EHWOR8 | EHWWOR8 | |

Figure 7-1 describes file sets A and B created and used by text search. These files are contained in library QUSRSYS.

Figure 7-1. Text Search Services Files and Member Names

| File | Member | Description |
|-------------|---------------|--|
| QABBADMTB | EHVADMTB | Administration table |
| QABBQTB | EHVIQTB | Document index queue |
| QABBCAN | EHVCANP1 | Candidates file – primary index A |
| QABBCAN | EHVCANP2 | Candidates file – primary index B |
| QABBCAN | EHVCANU1 | Candidates file – additional index A |
| QABBCAN | EHVCANU2 | Candidates file – additional index B |
| QABBCOX | EHVCOXP1 | Context index – primary index A |
| QABBCOX | EHVCOXP2 | Context index – primary index B |
| QABBCOX | EHVCOXU1 | Context index – additional index A |
| QABBCOX | EHVCOXU2 | Context index – additional index B |
| QABBDEX | EHVDEXP1 | External document identifiers – primary index A |
| QABBDEX | EHVDEXP2 | External document identifiers – primary index B |
| QABBDEX | EHVDEXU1 | External document identifiers – additional index A |
| QABBDEX | EHVDEXU2 | External document identifiers – additional index B |
| QABBDIC | EHVDICP1 | Dictionary – primary index A |
| QABBDIC | EHVDICP2 | Dictionary – primary index B |
| QABBDIC | EHVDICU1 | Dictionary – additional index A |
| QABBDIC | EHVDICU2 | Dictionary – additional index B |
| QABBDIX | EHVDIXP1 | Internal document identifiers – primary index A |
| QABBDIX | EHVDIXP2 | Internal document identifiers – primary index B |
| QABBDIX | EHVDIXU1 | Internal document identifiers – additional index A |
| QABBDIX | EHVDIXU2 | Internal document identifiers – additional index B |
| QABBDOX | EHVDOXP1 | Document index – primary index A |
| QABBDOX | EHVDOXP2 | Document index – primary index B |
| QABBDOX | EHVDOXU1 | Document index – additional index A |
| QABBDOX | EHVDOXU2 | Document index – additional index B |
| QABBFIX | EHVFIXP1 | Fragment index – primary index A |
| QABBFIX | EHVFIXP2 | Fragment index – primary index B |
| QABBFIX | EHVFIXU1 | Fragment index – additional index A |
| QABBFIX | EHVFIXU2 | Fragment index – additional index B |
| QABBLADN | EHWINDEX | List of LADNs indexed |

Backup and Recovery of Text Search Files

The following items are saved when you save text search files:

- The administration table
- The scheduling queue
- The primary index files and additional index files

These files should be backed up when the document library is backed up. See Chapter 9, “Data Backup, Recovery, and Storage Management” for information on how to back up the document library.

You should use the STRRGZIDX command before backing up your files. This removes deleted entries from the files. You may want to process the scheduling queue before backing up these files. This provides a faster recovery.

Language Dictionaries

Figure 7-2 lists the currently supported languages and language dictionaries. Language dictionaries corresponding to the languages in this table can be installed and used for indexing a document and for text search.

Figure 7-2 (Page 1 of 2). Languages and Language Dictionaries

| Language | Language Identifier | Dictionary | Dictionary Name |
|----------------------|----------------------------|-------------------------------------|------------------------|
| Belgian Dutch | NLB | Dutch Modern | AKTUEEL |
| Belgian French | FRB | French National | FRANCAIS |
| Brazilian Portuguese | PTB | Brazilian Portuguese | BRASIL |
| Canadian French | FRC | French Canadian | FRA2 |
| Catalan | CAT | Catalonian Spanish | CATALA |
| Danish | DAN | Danish | DANSK |
| Dutch (1) | NLD | Dutch - modern Dutch - preferred | AKTUEEL NEDERLND |
| French | FRA | French National | FRANCAIS |
| German | DEU | German | DEUTSCH |
| Icelandic | ISL | Icelandic | ISLENSK |
| Italian | ITA | Italian | ITALIANO |
| Norwegian Bokmål | NOR | Norwegian - Bokmål | NORBOK |
| Nynorsk Norwegian | NON | Norwegian - Nynorsk | NORNYN |
| Portuguese | PTG | Portuguese National | PORTUGAL |
| Spanish | ESP | Spanish | ESPANA |
| Swedish | SVE | Swedish | SVENSK |
| Swiss French | FRS | French National | FRANCAIS |
| Swiss German | DES | Swiss German | DSCHWEIZ |
| Swiss Italian | ITS | Italian | ITALIANO |
| U.K. English | ENG | U.K. English | UK |

Figure 7-2 (Page 2 of 2). Languages and Language Dictionaries

| Language | Language Identifier | Dictionary | Dictionary Name |
|--------------|---------------------|--------------|-----------------|
| U.S. English | ENU | U.S. English | US |

Note: (1) When the Dutch language identifier, NLD, is specified for a document, both language dictionaries for Dutch are used.

When using a language dictionary, linguistic processing stores key words or word parts into the text index. Linguistic processing also uses a **stop word list**, which identifies frequently used words. This reduces the amount of information stored in the text index. Also, text search can then locate documents containing related forms of a word.

A stop word list is a part of a national language dictionary that identifies words that are very common for the language. Words on the stop word list are not stored in the text index or used for text searches. Therefore, the text index uses less space on your system and text search processing takes less time because fewer words are used. Text searches are also more accurate because matches with common words in many or all documents are not made.

You can add words to or remove words from the stop word list for each language you use. For information about revising the stop word lists you use, see “Stop Word Lists” on page 7-13.

The document’s language ID, which is part of the document details, indicates the document’s language. If the language dictionary specified by the document’s language ID is not available when a document is indexed, all words of a document’s text are stored in the text index. In this case, text search cannot locate documents containing related forms of a word. If there is no language ID associated with a document, the system default language ID is used.

Linguistic Processing

Linguistic processing uses dictionaries that contain linguistic information about the words for a language and that include a stop word list. The dictionaries must be available for the languages that you want linguistically supported. The same linguistic processing is done for indexing documents and for query phrases, to ensure that the query phrase finds matches with documents that contain the query phrase.

If a language does not have a dictionary, or the dictionary is not installed, then the index update and search processing are done but do not include linguistic processing such as word-reduction or stop-word processing. Word separation and sentence separation are attempted, although language-specific processing is not possible. Also, the amount of linguistic processing done for one language can vary from other languages depending on the contents of the language dictionary.

The maximum linguistic processing used when a dictionary is available includes:

- Dividing the text into words and sentences.
- Filtering using the stop word list to discard commonly used words.
- Filtering based on the part of speech to keep only nouns, verbs, and adjectives as keywords. If the part of speech for a word is ambiguous, the word is kept.

- Word reduction to use the singular form of nouns, the infinitive form of verbs, and the base form of adjectives.
- Word decomposition for Germanic languages and words containing a hyphen or a slash, to use the component parts of the word.
- Word normalization to put all words into uppercase and remove diacritic marks from characters, if appropriate for the language.
- Synonym processing to return all synonyms for a word on a query.

The linguistic processing used when no dictionary is available includes:

- Dividing the text into words and sentences
- Word normalization

The following table shows the linguistic processing functions supported for each language:

Figure 7-3. Language Dictionary Contents

| Language | Stop Word Filtering | Part of Speech Filtering | Word Decomposition | Word Reduction | Synonyms | Notes |
|--------------------------------|---------------------|--------------------------|--------------------|----------------|----------|----------------------------|
| Belgian Dutch | Yes | Yes | Yes | Yes | Yes | Uses Dutch dictionary |
| Belgian French | Yes | Yes | No | Yes | Yes | Uses French dictionary |
| Belgian Portuguese | Yes | Yes | No | Yes | No | |
| Canadian French | Yes | Yes | No | Yes | Yes | |
| Catalan | Yes | Yes | No | Yes | Yes | |
| Danish | Yes | Yes | Yes | Yes | Yes | |
| Dutch Netherlands | Yes | Yes | Yes | Yes | Yes | |
| U.K. English | Yes | Yes | No | Yes | Yes | |
| U.S. English | Yes | Yes | No | Yes | Yes | |
| Finnish | No | No | No | No | No | No dictionary is available |
| French | Yes | Yes | No | Yes | Yes | |
| Swiss French | Yes | Yes | No | Yes | Yes | Uses French dictionary |
| German | Yes | Yes | Yes | Yes | Yes | |
| Swiss German | Yes | Yes | Yes | Yes | No | |
| Greek | No | No | No | No | No | No dictionary is available |
| Icelandic | Yes | No | Yes | No | No | |
| Italian | Yes | Yes | No | Yes | Yes | |
| Swiss Italian | Yes | Yes | No | Yes | Yes | Uses Italian dictionary |
| Norwegian, Bokmål | Yes | Yes | Yes | Yes | Yes | |
| Norwegian, Nynorsk | Yes | Yes | Yes | Yes | No | |
| Portuguese | Yes | Yes | No | Yes | Yes | |
| Spanish | Yes | Yes | No | Yes | Yes | |
| Swedish | Yes | Yes | Yes | Yes | Yes | |
| Turkish | No | No | No | No | No | No dictionary is available |
| Other Latin alphabet languages | No | No | No | No | No | No dictionary is available |

The following languages are not currently supported by text search:

- Japanese
- Korean
- Simplified Chinese

- Traditional Chinese
- Cyrillic alphabet languages

Special Processing

Linguistic processing performs special processing on words with slashes and hyphens, compound words, and uses generic characters to make searches easier to perform.

Words with Slashes: Words with slashes are processed as follows:

- A word containing a slash is used as a keyword.
For example, *administrators/supervisors* would be used as one keyword.
- The slashes are used as delimiters to return the slash word components as keywords.
In the example, *administrators* and *supervisors* are used as keywords.
- The slash word components receive full linguistic processing.
Linguistic processing such as changing *supervisors* to its base, singular form of *supervisor* is done as appropriate.
- The word with a slash, its base form, and each of the component words have the same relative position in the sentence. A component word may be discarded by stop-word or part-of-speech processing.

Words with Hyphens: Words with hyphens are processed as follows:

- A word containing required hyphens is used as a keyword.
- If the word is found in a dictionary, full linguistic processing is done for the word using the dictionary information available.
For example, *mother-in-law* for U.S. English receives full linguistic processing.
- If the word is not found in a dictionary, the hyphens are used as delimiters to find and return the components as keywords. The components of the word with hyphens get full linguistic processing.
For example, *CD-players* are used as one keyword, and *CD* and *players* are used as keywords. Linguistic processing such as changing *players* to its base, singular form of *player* is done as appropriate.
- The hyphenated word, its base form, and each of the component words have the same word number in the sentence. A word with hyphens or any of its component words may be discarded by stop-word or part-of-speech processing.

Words at the ends of lines that have been automatically hyphenated are recognized and rejoined. Spelling changes are made as appropriate to the word after hyphens are removed.

Compound Words: Compound words are processed as follows:

- For languages that allow compound words (for example, German), the entire compound word is used as a keyword.
- Component words are determined and processed linguistically. More than one decomposition for a compound word is possible.
- The base form for a compound word is formed by concatenating the first base form for each of the component words.

- If a word is found as a whole in a dictionary, then no compound word processing is done. The word, although it is a compound word, is treated as a single word and component words are not assumed to have valid independent meaning.
- The compound word and each of its component words has the same relative position in the sentence. A component word may be discarded by stop-word or part-of-speech processing.

Combinations of Special Processing: Words with hyphens or slashes and compound words are processed in any combination. A word can contain slashes, hyphens, and a compound word component.

Multiple Base Forms: Linguistic processing can use multiple base forms (verbs, nouns, and adjectives) for a word, and text search uses all of the unique base forms as keywords. If two base forms have the same spelling, duplicate forms are discarded.

Generic Character Processing: An asterisk (*) or a question mark (?) can be used as part of a word when specifying a search or query phrase. An asterisk (*) represents one or more alphanumeric characters and a question mark (?) represents one alphanumeric character.

Linguistic processing is not done for search terms that contain an asterisk or question mark. For example, *men in a search is not processed to find the base form for men.

Language IDs

The language ID of a document must be set for linguistic processing to use the correct language dictionary when indexing the document. Linguistic processing helps you find more documents during a search than if the documents had been indexed without linguistic processing.

For example, you may have documents that contain several forms of the word *program*, such as *programs*, *programming*, and *programmed*. If the document is indexed without linguistic processing, a search for documents with the word *program* would find only those documents with the word *program*. With linguistic processing, the search finds documents that contain all forms of the word.

Documents created before OfficeVision/400 Version 2 Release 1 Modification 1 may not have a language ID. Documents from other office products also may or may not have a language ID. The system value, QLANGID, is used to determine the language for documents that do not have a language ID.

Setting Language IDs to Groups of Documents

You can set the language ID in one of several ways:

- Set all documents in a folder to the same language ID.
- Set all documents that belong to a user to the same language ID.
- Set all the documents on your system to the same language ID. This is possible if only one language is used on your system. You may want to do this first if almost all of your documents are in the same language. This way you can easily set all the documents and then go back and change the few that are not in that language.

The country ID must be set when the language ID is set, if it is not already set for a document. Figure 7-2 on page 7-7 shows the language IDs and dictionaries supported by text search services. To see a list of country IDs, view the help information for the Change Document Details (CHGDOCD) command.

You can use the QRYDOCLIB command to retrieve a list of all the documents on the system, and submit indexing requests using the ADDTXTIDX command. “Change Document Language (ZCHGDOCLNG) Command Example” on page 7-19 describes how to change the language ID or country ID for a list of documents. “Index Documents (ZIDXDOC) Command Example” on page 7-24 describes how to index a list of documents created with the QRYDOCLIB command.

In a program, you can decide to index all documents with a specific type, for example type RFTDCA, RFTAS400, FFTDCA, or FFTAS400. This may include documents with those types shipped in folders and supplied with products you have installed on your system. You can index any document type that has a document conversion program to convert it to type RFTDCA or FFTDCA. Documents of any other type, such as PCFILE (with no appropriate document conversion program), cannot be indexed and a request to index documents with other types does not complete. If you have installed products that have folders, you can examine these folders for documents with a type that can be indexed. If you do not want these folders indexed, you can remove these documents from the text index using the RMVTXTIDX command, specifying the folders to remove. (When reviewing folders, IBM-supplied folders start with the letter Q.)

Stop Word Lists

A stop word list is a part of a national language dictionary that identifies words that are very common in the language. Words on the stop word list are not stored in the text index or used for text searches.

You can create a user stop word list to add words to or remove words from the stop word list for each language you use. You use the Create Spelling Aid Dictionary (CRTSPADCT) command to create the stop word list. For more information, see “Create Spelling Aid Dictionary (CRTSPADCT) Command” on page 8-17.

This section describes how to review the stop word list provided with the dictionaries you use and how to change the stop word list used for a language.

If you want to see the stop word list for a dictionary, use the Print Stop Word List (PRTSWL) CL command to print the list. To print the IBM-supplied stop word list for U.S. English, use the following command:

```
PRTSWL LANGID(ENU) TYPE(*IBM)
```

Use the Retrieve Stop Word List Source (RTVSWLSRC) command to retrieve the words in an existing stop word list into a source file. For example, to retrieve the IBM-supplied stop word list source for U.K. English, use the following command:

```
RTVSWLSRC LANGID(ENG) SRCFILE(MYLIB/MYFILE) SRCMBR(MYMBR)
```

You can create a source file and edit it to include the words you want added or removed from the IBM-supplied stop word list. Or, you can retrieve the IBM-supplied stop word list into a source file and edit the new input source file as preparation for creating a user stop word list. Follow these guidelines when creating an input source file:

- The first record should contain a keyboard ID in the first 3 characters.
- The last record should contain a CHRID (see “Character Set and Code Page Considerations” on page 8-19).
- Use one word per record to add to the stop word list.
- Use records with words preceded by one or more dashes (–) to prevent the word from being added to the user-created stop word list from the IBM-supplied stop word list.
- Add lines beginning with dot commands or asterisks (*), if desired, which are ignored.
- The CCSID of the source file should be *HEX.

The following example shows a source file for a stop word list that adds the term IBM and removes the word begin from the stop word list.

```
* Common terms in our documents
IBM
* Do not stop BEGIN because it is also a person's name *
-begin
.chrid 037
```

You can add words or abbreviations that are commonly used in your office but are not useful for locating documents when searching. For example, if engineering work is done in your office, all documents except letters might contain the word *specification*, which would not help users locate documents. Or, you can use a dictionary that does not include a stop word list, and want to create a stop word list.

If you want to use the input source file you created to create a user stop word list, use the CRTSPADCT CL command.

The following command creates a stop word list in library QUSRSYS. The words used in the list are from source member ENU in the source file SRC in library MYLIB.

```
CRTSPADCT SPADCT(*USRSWL) SRCFILE(MYLIB/SRC) SRCMBR(*DCT)
      OPTION(*SRC) BASESWL(*IBM) SWLLANGID(ENU) REPLACE(*N0)
```

The new stop word list is named QTWENUX and has a type of *SPADCT. The new stop word list includes the words in the IBM-supplied stop word list for U.S. English. If there is an existing stop word list in library QUSRSYS with the same name, the existing stop word list is not replaced and the command does not complete.

In addition to creating a new stop word list, the command changes the stop word list that is used during text search processing. When text search processes indexing requests, it uses the stop word list you create, if one exists. Otherwise, the IBM-supplied stop word list is used. For more information on the CRTSPADCT command, see “Create Spelling Aid Dictionary (CRTSPADCT) Command” on page 8-17.

Text Search Services CL Commands

Text search provides several commands that you can enter on the command line on any AS/400 system display. The following commands allow you to directly control text search administrative functions:

- Add Text Index Entry (ADDTXTIDX)
- Display Text Index Status (DSPIDXSTS)
- End Index Monitor (ENDIDXMON)
- Remove Text Index Entry (RMVTEXTIDX)
- Start Index Monitor (STRIDXMON)
- Start Reorganization of Index (STRRGZIDX)
- Start Update of Index (STRUPIDX)
- Work with Text Index (WRKTEXTIDX)

Note: For complete descriptions of these commands, see Appendix D, “Control Language (CL) Commands.”

Add Text Index Entry (ADDTXTIDX) Command

Use the Add Text Index Entry (ADDTXTIDX) command to add index entries to the text index for one or more documents. Documents are indexed during the next index update.

You can index a single document by specifying any of the following on the command:

- The document-in-folder (DOC) name and the folder (FLR) in which the document is located
- The library-assigned document name (LADN) using DOC(*DOCID) and the DOCID parameter
- The system object name using DOC(*SYSOBJNAM) and the SYSOBJNAM parameter

To index all documents in a folder, specify DOC(*ALL) and the folder in the FLR parameter.

Use the IDXTXT parameter to determine whether the document should be indexed unconditionally or indexed only if it is currently in the text search index. IDXTXT(*PREVIOUS) allows you to conveniently index again all the objects in a folder that are currently part of the index.

The USRID parameter specifies the user ID and address of the user indexing the document. The user ID and address must be enrolled in the system distribution directory. If *CURRENT is specified, then the user associated with the job is the user making the request. If a user ID and address (other than that of the user associated with the job) is specified, then the current user must be able to work on behalf of the specified user before the request can be made to index the document. The authority to the document is checked for the user on whose behalf the job's user is working and not for the job's user. Use the Grant User Permission (GRTUSRPMN) command to give users the ability to work on behalf of other users.

Security Considerations

You, or the user on whose behalf the ADDTXTIDX command is being issued, must have at least *USE authority to each document to be indexed or have *ALLOBJ special authority. Authority to a folder is not required when the FLR parameter is specified.

Note: Access authority to the ADDTXTIDX command is used to determine if individual users have authority to index documents from OfficeVision/400 displays.

Some Ways to Use This Command

To create text index entries by document, type:

```
ADDTXTIDX
```

You are prompted to select the documents you want to add to the text index, and then your requested documents are added to the scheduling queue.

Character Set and Code Page Considerations

The CMDCHRID parameter allows you to specify the character set and code page in which the characters are entered in the USRID parameter. The USRID is translated from the CMDCHRID to character set 930, code page 500 so that it can be verified as existing in the system distribution directory and the ability to work on behalf of another user can be verified. *SYSVAL means that the character set and code page are taken from the QCHRID system value. *DEVD means that the character set and code page are taken from the display device. *DEVD is valid only from an interactive job. An error message is returned if *DEVD is used in a batch job.

Display Text Index Status (DSPIDXSTS) Command

Use the Display Text Index Status (DSPIDXSTS) command to display index status information such as the last update, next update, or number of documents on the scheduling queue.

This command is shipped with public *USE authority.

End Index Monitor (ENDIDXMON) Command

Use the End Index Monitor (ENDIDXMON) command to end the automatic update or reorganization of the index (stop the administration monitor) based on the values specified in the index details. A request is sent to stop administration monitor at the beginning of the next polling cycle.

This command is shipped with public *EXCLUDE authority.

Remove Text Index Entry (RMVTEXTIDXE) Command

Use the Remove Text Index Entry (RMVTEXTIDXE) command to remove index entries from the text index for one or more documents. Index entries are removed during the next index update.

To remove a single document, specify any of the following:

- The document-in-folder (DOC) name and the folder (FLR) in which the document is located
- The library-assigned document name (LADN) using DOC(*DOCID) and the DOCID parameter
- The system object name using DOC(*SYSOBJNAM) and the SYSOBJNAM parameter

To remove all documents in a folder specify DOC(*ALL) and the folder in the FLR parameter.

The USRID parameter specifies the user ID and address of the user removing the document. The user ID and address must be enrolled in the system distribution directory. If *CURRENT is specified, then the user associated with the job is the user making the request. If a user ID and address other than that of the user associated with the job is specified, then the current user must be able to work on behalf of the specified user before the request can be made to remove the document. The authority to the document is checked for the user on whose behalf the job's user is working and not for the job's user. Use the Grant User Permission (GRTUSRPMN) command to give users the ability to work on behalf of other users.

Security Considerations

You, or the user on whose behalf the RMVTEXTIDXE command is being issued, must have at least *USE authority to each document to be removed or have *ALLOBJ authority. Authority to a folder is not required when the FLR parameter is specified. This command is shipped with public *EXCLUDE authority.

Some Ways to Use This Command

To remove text index entries by document, type:

```
RMVTEXTIDXE
```

You are prompted to select the documents you want to remove from the text index, and then the requested documents are removed from the scheduling queue.

Character Set and Code Page Considerations

The CMDCHRID parameter allows you to specify the character set and code page in which the characters are entered in the USRID. The USRID is translated from the CMDCHRID to character set 930, code page 500 so that it can be verified as existing in the system distribution directory and the ability to work on behalf of another user can be verified. *SYSVAL means that the character set and code page are taken from the QCHRID system value. *DEV D means that the character set and code page are taken from the display device. *DEV D is valid only from an interactive job. An error message is returned if *DEV D is used in a batch job.

Start Index Monitor (STRIDXMON) Command

Use the Start Index Monitor (STRIDXMON) command to activate the automatic update or reorganization of the index based on the values specified in the index details.

This command is shipped with public *EXCLUDE authority.

Start Reorganization of Index (STRRGZIDX) Command

Use the Start Reorganization of Index (STRRGZIDX) command to start reorganization of the index.

This command is shipped with public *EXCLUDE authority.

Text searches are allowed during the text index reorganization.

When commands are issued during a text index reorganization, the following occurs:

- If the STRUPDIDX or STRRGZIDX command is entered during a text index reorganization, a message is returned stating that this command cannot be used at this time.
- If the STRIDXMON or ENDIDXMON command is entered during a text index reorganization, the process completes before the command takes effect.
- If the ADDTXTIDX or RMVTXTIDX command is entered during a text index reorganization, the command completes and its requests are placed on the scheduling queue.
- If the QRYDOCLIB command is entered and includes text search criteria, the command completes and the results of the search are returned. This is also true for searches done through OfficeVision/400.

Start Update of Index (STRUPDIDX) Command

Use the Start Update of Index (STRUPDIDX) command to start processing the index queue.

This command is shipped with public *EXCLUDE authority.

Text searches are available during the text index updating process.

When commands are issued during a text index update the following occurs:

- If the STRUPDIDX or STRRGZIDX command is entered during a text index update, a message is returned stating that this command cannot be used at this time.
- If the STRIDXMON or ENDIDXMON command is entered during a text index update, the process completes before the command takes effect.
- If the ADDTXTIDX or RMVTXTIDX command is entered during a text index update, the command completes and its requests are placed on the scheduling queue.
- If the QRYDOCLIB command is entered and includes text search criteria, the command completes and the results of the search are returned. This is also true for searches done through OfficeVision/400.

Work with Text Index (WRKTXIDX) Command

Use the Work with Text Index (WRKTXIDX) command to display the administration panels. Use these panels to view the current status of the index and to change the attribute values that control the automatic update operations of the index.

This command is shipped with public *EXCLUDE authority.

Change Document Language (ZCHGDOCLNG) Command Example

Documents that are to be indexed are processed based on the language identifier associated with the document. If the document does not have a language identifier, the system language identifier value (QLANGID) is used. If this is not the correct identifier for a document, you can change the language identifier using the CHGDOCD command.

The examples in this section show the command source and command processing program used to create a command that can process a list of documents created using the QRYDOCLIB command. The name of the command is ZCHGDOCLNG and it is available in the QUSRTOOL library. See the instructions in the T9758INF member in the QATTINFO file in the QUSRTOOL library for information on how to create this example.

Use the ZCHGDOCLNG command to change the language ID or the country ID for a single document (either by specifying the name of the document if it is in a folder or by specifying the document identifier for the document). The command can be used to change a list of documents to the same language ID or country ID. Create the list by using the QRYDOCLIB command to place the list into an outfile. The description record is the only type of record that should appear in the QRYDOCLIB outfile. All documents have a description, so it is always guaranteed that a record appears in the outfile if the search values are met.

The language ID and country ID must be set to a value that is allowed by the system. To see a list of valid values, type CHGDOCD on a command line and press the PF4 key to see a prompt on the language ID or country ID fields.

Note: If a document does not have a language ID and country ID (for example, documents created before Version 2 Release 1 Modification 1 do not have this information), both the language ID and country ID must be specified when changing the document. If the document already has this information, then either the language ID or country ID or both can be changed.

The following table describes the ZCHGDOCLNG command parameters.

Figure 7-4. ZCHGDOCLNG Parameters

| Parameter | How to Use |
|---------------------|---|
| Document | Use this field to specify which document or documents are to be changed. <ul style="list-style-type: none"> • Name: When a user-defined-name is used, the name of the folder that the document is in must also be specified. • *INFILE: When *INFILE is specified, the input file name must be specified. • *DOCID: When *DOCID is specified, the document identifier must be specified. |
| Folder | Use this field to specify the folder, if any, that the document is in. Set to *NONE if the document parameter is set to *INFILE or *DOCID. |
| Language identifier | Use this field to specify the new language ID for the documents. Set to *SAME if you do not want the language ID changed. |
| Country identifier | Use this field to specify the new country ID for the documents. Set to *SAME if you do not want the country ID changed. |
| Input file name | Use this field to specify the name of the file that contains the results from running the QRYDOCLIB command. Only fill in this field if the document field is set to *INFILE. |
| Library name | Use this field to specify the name of the library that contains the input file. |
| Input member | Use this field to specify the name of the member that contains the results of the QRYDOCLIB command. If the first member of the file contains the result, *FIRST can be used for this field. |
| Document identifier | Use this field when you want to specify a document using the document identifier. |

Example 1: Change language ID to Belgium Dutch (NLB) for a document that is in a folder

Use the following command to change the language ID:

```
ZCHGDOCLNG DOC(MYDOC) FLR(MYFLR) LANGID(NLB)
```

Example 2: Change country ID to the Netherlands (NL) for a document that is not in a folder

Use the following command to change the country ID:

```
ZCHGDOCLNG DOC(*DOCID) LANGID(NL) DOCID(1991121219933838SYSTEM01)
```

Example 3: Change language ID and country ID for all documents that USER1 owns

Use the following to change the language ID and country ID for a list of documents created with the QRYDOCLIB command:

```
QRYDOCLIB QRYDFN(*IF ((*OWNER *EQ USER1)))  
          DOCL(*NONE) OUTFILE(MYLIB/MYFILE)  
ZCHGDOCLNG DOC(*INFILE) LANGID(NLB) CNTRYID(NL)  
          INFILE(MYLIB/MYFILE)
```

The following example shows the command source for the ZCHGDOCLNG command.

```

/*****
/*
/* Part name:   T9758CHS
/*
/* Source file: QATTCMD in library QUSRTOOL
/*
/* Function:    This is the command source for the ZCHGDOCLNG
/*              command.
/*
/*
/*****
      CMD      PROMPT('Change Document Language')
      PARM     KWD(DOC) TYPE(*CHAR) LEN(12) +
              SPCVAL((*INFILE) (*DOCID)) MIN(1) +
              CHOICE('Name, *INFILE, *DOCID') +
              PROMPT('Document')
      PARM     KWD(FLR) TYPE(*CHAR) LEN(63) DFT(*NONE) +
              SPCVAL((*NONE)) PROMPT('Folder')
      PARM     KWD(LANGID) TYPE(*CHAR) LEN(5) DFT(*SAME) +
              SPCVAL((*SAME)) PROMPT('Language identifier')
      PARM     KWD(CNTRYID) TYPE(*CHAR) LEN(5) DFT(*SAME) +
              SPCVAL((*SAME)) PROMPT('Country identifier')
      PARM     KWD(INFILE) TYPE(Q2) MIN(0) ALWVAR(*YES) +
              FILE(*IN) PMTCTL(P3) PROMPT('Input file +
              name')
      PARM     KWD(INMBR) TYPE(*SNAME) LEN(10) DFT(*FIRST) +
              SPCVAL((*FIRST)) EXPR(*YES) PMTCTL(P3) +
              PROMPT('Input member')
      PARM     KWD(DOCID) TYPE(*CHAR) LEN(24) RSTD(*NO) +
              DFT(*NONE) SPCVAL((*NONE)) MIN(0) MAX(1) +
              FILE(*NO) FULL(*NO) PASSATR(*NO) +
              PMTCTL(P1) PROMPT('Document identifier')
P1:          PMTCTL  CTL(DOC) COND((*EQ *DOCID))
P3:          PMTCTL  CTL(DOC) COND((*EQ *INFILE))
Q2:          QUAL    TYPE(*NAME) LEN(10) MIN(1) EXPR(*YES)
              QUAL    TYPE(*NAME) LEN(10) DFT(*LIBL) +
              SPCVAL((*LIBL) (*CURLIB *CURLIB)) +
              EXPR(*YES) PROMPT('Library name')

```

Figure 7-5. Command Source for the ZCHGDOCLNG Command

The following example shows the command processing program for the ZCHGDOCLNG command.

```

/*****
/*
/* Part name:   T9758CHP
/*
/* Source file: QATTCL in library QUSRT0OL
/*
/* Function:    This is the command processing program for the
/*              ZCHGDOCLNG command.
/*
/*
*****/
PGM          PARM(&DOC &FLR &LANGID &CNTRYID &INFILE +
              &INMBR &DOCID)
/*****
/* Declare program parameters.
*****/
DCL          VAR(&DOC) TYPE(*CHAR) LEN(12)
DCL          VAR(&FLR) TYPE(*CHAR) LEN(63)
DCL          VAR(&LANGID) TYPE(*CHAR) LEN(3)
DCL          VAR(&CNTRYID) TYPE(*CHAR) LEN(2)
DCL          VAR(&INFILE) TYPE(*CHAR) LEN(20)
DCL          VAR(&INMBR) TYPE(*CHAR) LEN(10)
DCL          VAR(&DOCID) TYPE(*CHAR) LEN(24)
/*****
/* Declare program variables.
*****/
DCL          VAR(&IFILE) TYPE(*CHAR) LEN(10) +
              VALUE('*INFILE ')
DCL          VAR(&INNAME) TYPE(*CHAR) LEN(10)
DCL          VAR(&INLIB) TYPE(*CHAR) LEN(10)
/*****
/* Declare input file format.
*****/
DCLF        FILE(QSYS/QAOSIQDL) RCDfmt(OSQDL)
/*****
/* Send an abnormally terminated message if any problems occur.
*****/
MONMSG      MSGID(CPF0000) EXEC(GOTO CMDLBL(TERM))
/*****
/* Single document processing.
*****/
IF          COND(&DOC *EQ &IFILE) THEN(GOTO +
              CMDLBL(INFILE)) /* If not *INFILE then +
              process a single document */
CHGDOCD     DOC(&DOC) FLR(&FLR) DOCID(&DOCID) +
              DOCLANGID(&LANGID) DOCCNTRYID(&CNTRYID) +
              /* Process flr/doc or *DOCID */
GOTO        CMDLBL(EOP)

```

Figure 7-6 (Part 1 of 2). Command Processing Program for the ZCHGDOCLNG Command

```

/*****
/* Input file processing.
/*****
INFILE:
      CHGVAR      VAR(&INLIB) VALUE(%SUBSTRING(&INFILE 11 10))
      CHGVAR      VAR(&INNAME) VALUE(%SUBSTRING(&INFILE 1 10))
      OVRDBF      FILE(QAOSIQDL) TOFILE(&INLIB/&INNAME) +
                  MBR(&INMBR)
      OPNDBF      FILE(QAOSIQDL) OPTION(*INP)
READF:  RCVF      RCDfmt(OSQDL)
      MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
      CHGDOCD     DOC(*DOCID) DOCID(&QDLID) +
                  DOCLANGID(&LANGID) DOCCNTRYID(&CNTRYID) +
                  /* Using document identifier, change +
                  document to new language and country +
                  identifier.
                  */
      GOTO        CMDLBL(READF)
TERM:   SNDPGMMSG MSG('Command terminated abnormally. See +
                  previous messages.')
EOF:    CLOF      OPNID(QAOSIQDL)
      MONMSG      MSGID(CPF0000)
EOP:    ENDPGM

```

Figure 7-6 (Part 2 of 2). Command Processing Program for the ZCHGDOCLNG Command

Index Documents (ZIDXDOC) Command Example

The examples in this section show the command source and command processing program to create a command that can index a list of documents created using the QRYDOCLIB command. The name of the command is ZIDXDOC and it is available in the QUSRTOOL library. See the instructions in the T9758INF member in the QATTINFO file in the QUSRTOOL library for information on how to create this example.

Use the ZIDXDOC command to index a list of documents. Create the list by using the QRYDOCLIB command to place the list into an outfile. The description record is the only record that should appear in the QRYDOCLIB outfile. All documents have a description, so it is always guaranteed that a record appears in the outfile if the search values are met. When this command is complete, start the indexing process if it is not automatic or already running. Do this using the STRUPDIDX command.

The following table describes the ZIDXDOC command parameters.

Figure 7-7 (Page 1 of 2). ZIDXDOC Parameters

| Parameter | How to Use |
|-----------------|--|
| Input file name | Use this field to specify the name of the file that contains the results from running the QRYDOCLIB command. |
| Library name | Use this field to specify the name of the library that contains the input file. |

Figure 7-7 (Page 2 of 2). ZIDXDOC Parameters

| Parameter | How to Use |
|------------------|--|
| Input member | Use this field to specify the name of the member that contains the results of the QRYDOCLIB command. If the first member of the file contains the result, *FIRST can be used for this field. |

Example: Index all documents that you own

Use the following commands to create a list of all the documents you own and start the indexing process:

```
QRYDOCLIB QRYDFN(*IF ((*OWNER *EQ USER1)))
          DOCL(*NONE) OUTFILE(MYLIB/MYFILE)
ZIDXDOC INFILE(MYLIB/MYFILE)
STRUPDIDX
```

The following example shows the command source for the ZIDXDOC command.

```
/*
/*
/* Part name:   T9758IXS
/*
/* Source file: QATTCMD in library QUSRT00L
/*
/* Function:    This is the command source for the ZIDXDOC command.
/*
/*
/*****
          CMD          PROMPT('Index Documents')
          PARM          KWD(INFILE) TYPE(Q2) MIN(1) ALWVAR(*YES) +
                       FILE(*IN) PROMPT('Input file name')
          PARM          KWD(INMBR) TYPE(*SNAME) LEN(10) DFT(*FIRST) +
                       SPCVAL((*FIRST)) EXPR(*YES) PROMPT('Input +
                       member')
Q2:          QUAL          TYPE(*NAME) LEN(10) MIN(1) EXPR(*YES)
          QUAL          TYPE(*NAME) LEN(10) DFT(*LIBL) +
                       SPCVAL((*LIBL) (*CURLIB *CURLIB)) +
                       EXPR(*YES) PROMPT('Library name')
*****/
```

Figure 7-8. Command Source for the ZIDXDOC Command

The following example shows the command processing program for the ZIDXDOC command.

```

/*****
/*
/* Part name:   T9758IXP
/*
/* Source file: QATTCL in library QUSRTOOL
/*
/* Function:    This is the command processing program for the
/*              ZIDXDOC command.
/*
/*****
PGM          PARM(&INFILE &INMBR)
/*****
/* Declare program parameters.
/*****
DCL          VAR(&INFILE) TYPE(*CHAR) LEN(20)
DCL          VAR(&INMBR) TYPE(*CHAR) LEN(10)
/*****
/* Declare program variables.
/*****
DCL          VAR(&IFILE) TYPE(*CHAR) LEN(10) +
              VALUE('*INFILE ')
DCL          VAR(&INNAME) TYPE(*CHAR) LEN(10)
DCL          VAR(&INLIB) TYPE(*CHAR) LEN(10)
/*****
/* Declare input file format.
/*****
DCLF        FILE(QSYS/QAOSIQDL) RCDFMT(OSQDL)
/*****
/* Send an abnormally terminated message if any problems occur.
/*****
MONMSG      MSGID(CPF0000) EXEC(GOTO CMDLBL(TERM))
MONMSG      MSGID(OF0000) EXEC(GOTO CMDLBL(TERM))
/*****
/* Generate requests to index all the documents in the input file.
/* It is possible for the request to be rejected, see the messages
/* generated for the cause.
/*****
CHGVAR      VAR(&INLIB) VALUE(%SUBSTRING(&INFILE 11 10))
CHGVAR      VAR(&INNAME) VALUE(%SUBSTRING(&INFILE 1 10))
OVRDBF      FILE(QAOSIQDL) TOFILE(&INLIB/&INNAME) +
              MBR(&INMBR)
OPNDBF      FILE(QAOSIQDL) OPTION(*INP)
READF:      RCVF          RCDFMT(OSQDL)
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))
ADDTXTIDXE  DOC(*DOCID) DOCID(&QDLID) /* Index the +
              document using the document identifier +
              since all documents have one. Not all +
              documents are in a folder. */
GOTO        CMDLBL(READF)
TERM:       SNDPGMMSG    MSG('Command terminated abnormally. See +
              previous messages.')
EOF:        CLOF         OPNID(QAOSIQDL)
              MONMSG      MSGID(CPF0000)
EOP:        ENDPGM

```

Figure 7-9. Command Processing Program for the ZIDXDOC Command

Chapter 8. Word Processing Services

This chapter describes word processing services, discusses objects, and tells how to create a help document. It also lists the word processing services application and control language (CL) interfaces, provides programming examples, and discusses output file considerations.

Note: This chapter assumes that the OfficeVision/400 word processing function is always used to work with documents. If other office applications have been defined using the OfficeVision/400 application enabler application program interfaces (APIs), those programs are called in place of the OfficeVision/400 word processing function. For more information about defining other office applications, see "OfficeVision/400 V2R3 Application Enabler Support" on page 11-12.

Word Processing Services Overview

This section provides a description of the word processing services. The overview includes information on document functions, printing and merging text and data, dictionaries, and text profiles.

Document Functions

Several document functions are available through the word processing services of the AS/400 system. You can easily create, change, copy, view, and check spelling and grade-level word use without having to use the menus of OfficeVision/400. These document functions are performed using the CL commands listed below. You can access the following functions by selecting the CL commands from a user-defined menu, or using them as part of a CL program:

| | |
|------------------|---|
| CHKDOC | Check the spelling or grade-level of a document |
| CPYDOC | Copy a document |
| CRTDOC | Create a document |
| DSPHLPDOC | Display a help text document |
| DSPDOC | Display (view) a document |
| EDTDOC | Edit (revise) a document |
| FILLFORM | Fill in a form field document |
| WRKDOC | Work with documents |
| WRKFLR | Work with folders |

Printing and Merging Text and Data

You can also paginate, print, and merge data and text to produce group mailings using word processing services. You may merge text with data from a file, a query, or another document. The merged information can be used to produce printed output, a database file, or another document that can be changed. You can also work with the documents waiting to be printed. You can access these functions by selecting an option from a user-defined menu, or as part of a CL program. Printing functions are performed using the following CL commands:

| | |
|-------------------|------------------------------------|
| MRGDOC | Merge a document |
| PAGDOC | Paginate a document |
| PRTDOC | Print a document |
| WRKDOCPRQT | Work with the document print queue |

Dictionaries

Dictionaries can be created or deleted easily using word processing services. You can also perform spell or grade level checking in the background, allowing you to perform other tasks while the document is checked.

The dictionary functions are performed using the CL commands listed below. These functions can be called by selecting an option from a user-defined menu, or as part of a CL program:

CRTSPADCT Create a spelling aid dictionary
DLTSPADCT Delete a spelling aid dictionary

Text Profiles

You may create, change, copy, delete, or activate text profiles using word processing services. Text profile functions are performed from the Work with Text Profiles display, that can be accessed using word processing services.

The text profile functions are performed using the CL command, Work with Text Profiles (WRKTXTPRF).

You can call this function by selecting an option from a user-defined menu, or as part of a CL program.

Word Processing Services Objects

This section describes the word processing services objects, including text profiles, dictionaries, database files, notepads, help text documents, and queries. Refer to “Documents” on page 5-5 for more information on documents. Refer to “Folders and Document Lists” on page 5-6 for more information on folders.

Text Profiles

Text profiles are documents stored in an IBM-supplied folder (QPRFFLR). Text profiles contain the options that define how a document will look when it is printed. Each time a new document is created, the currently active text profile document is copied to the new document. Therefore, the new document receives all the same formatting options, print options, and so on as the text profile document.

The folder QPRFFLR is shipped with one text profile document called SYSTEM, which is referred to as the *system profile*. This text profile can be changed to fit the user’s system, but it cannot be deleted. Until other text profiles are activated or assigned to folders, the system profile is used when new documents are created. All user-created text profiles use the system profile as a base.

A user can create text profiles that define desired defaults for a specific type of document, such as an annual report or a meeting notice. Text profiles can be activated for a specific user, or they can be assigned to a folder so that all documents created in that folder receive the same defaults regardless of which user creates the document. Text profiles can be assigned to a folder when the folder is created (Create Folder display), or when changing a folder’s details (Change Folder Details display).

Since text profiles are actually documents, they may contain text as well. To change a folder document, use the EDTDOC command, or select option 2 (Revise) from the Work with Documents in Folders display, while working with folder QPRFFLR. All text profiles contain the following options:

- Document format
- Alternate format
- Dictionary
- Print (both normal and label)
- Merge (used by the MRGDOC command)
- Numeric field editing
- Various edit
- Text month names

Text profile functions can be accessed in the following ways:

- WRKTXTPRF command.
- Press F9 on the Work with... display in OfficeVision/400 word processing. Then select option 5 from the Work with Word Processing menu.
- Select option 6 on STROFC or STRWP, then select option 5 from the Work with Word Processing menu.

Dictionaryes

Language dictionaries on the AS/400 system are unique objects of type *SPADCT. These objects reside in libraries. Some language dictionaries can be purchased with the system and reside in library QDCT. The system automatically creates user dictionaries when you use the Add Word function from within an edit session. You can also create user dictionaries with CL commands.

IBM language dictionaries are purchased as a group. These language dictionaries reside in library QDCT. The language dictionaries supported are listed below:

| Language | Dictionary |
|--------------------------|------------|
| Brazilian Portuguese | (BRASIL) |
| Catalonian Spanish | (CATALA) |
| Danish | (DANSK) |
| Dutch — Modern | (AKTUEEL) |
| Dutch — Preferred | (NEDERLND) |
| English — American | (US) |
| English — United Kingdom | (UK) |
| English — Legal | (LEGAL) |
| English — Medical | (MEDICAL) |
| French | (FRANCAIS) |
| French — Canadian | (FRA2) |
| Finnish | (SUOMI) |
| German | (DEUTSCH) |
| Greek | (GREEK) |
| Icelandic | (ISLENSK) |
| Italian | (ITALIANO) |
| National Portuguese | (PORTUGAL) |
| Norwegian — Bokmål | (NORBOK) |
| Norwegian — Nynorsk | (NORNYN) |
| South African | (AFRIKAAN) |
| Spanish | (ESPANA) |
| Swedish | (SVENSK) |
| Swiss-German | (DSCHWEIZ) |

User Dictionaries

User dictionaries store words that are not contained in an IBM language dictionary, but are used often by a particular department, work group, or user. Proper names, code names, and other terms unique to the user's business are added to a user dictionary so they are not highlighted as errors when documents are checked for spelling.

User dictionaries and stop word lists can be created using the Create Spelling Aid Dictionary (CRTSPADCT) command. User dictionaries do not need a specific file and member name. They can reside in any user library. User-created stop word lists are stored in the QUSRSYS library with names of QTW nnn X, where nnn is the language ID.

Input for the CRTSPADCT command is a source file member that contains the words to be included in the dictionary or stop word list. The words in this source file can be entered in one of two ways:

- One word per line
- Several words per line, each separated by a space

You can define hyphenation points in the words you add to your source file by inserting hyphens between the syllables (dic-tion-ar-y). For words that contain required hyphens, such as mother-in-law, two hyphens need to be entered at these locations (moth-er--in--law).

Dictionaries to be checked against are entered on the Dictionary Options display.

Database Files

Database files can be created by printing a document to a file using the PRTDOC command, or by writing a folder's contents to a file using the DSPFLR command.

Printing a Document to a File Using the PRTDOC Command

You can write the contents of an AS/400 document to a database file by using the PRTDOC command, or by specifying option 3 (File) for the *Output device* prompt on the Print Options display. You specify the name of the file, library, and member to receive the printed document. If the file does not already exist, it is created. If the file already exists, the printed output is added to the end of the specified file member. If the file is in use by another user, an error message is returned. Since the print options are part of a text profile, a text profile may specify a database file as the output file. Each document created using this text profile is written automatically to the specified database file when printed. See Appendix C, "Office Services Output File Considerations," for the format of the PRTDOC output file.

Files Produced Using the DSPFLR Command

A database file containing a list of a folder's contents (either documents or folders) can be created using the DSPFLR command with the value *OUTFILE specified for the OUTPUT parameter. The file is created if it does not already exist. The file member contains either a list of folders within the specified folder, or a list of documents within the specified folder. Refer to Appendix C, "Office Services Output File Considerations," for more information about the output file layout for the DSPFLR command.

Folder Object Information: The model output file supplied with the system is named QADSPFLR in library QSYS and is used when the following command is specified:

```
DSPFLR TYPE(*FLR) OUTPUT(*OUTFILE)
```

See Appendix C, "Office Services Output File Considerations," for more information about the model output file.

Document Object Information: The other model output file supplied with the system is named QADSPDOC in library QSYS and is used when the following command is specified:

```
DSPFLR TYPE(*DOC) OUTPUT(*OUTFILE)
```

Refer to Appendix C, "Office Services Output File Considerations," for more information about the model output file.

Notepads

Notepads in OfficeVision/400 word processing services are documents. Notepads are used to store and retrieve text that is used often. Text can easily be copied or moved to a notepad, and recalled from the notepad.

A system-named notepad document is created for each user when the notepad function is first accessed. This notepad document is given the same name as the user's ID, and is stored in the IBM-supplied folder QWPDOCS. If you need to use more than one notepad, specify user-named notepads on the Change Editing Options display (F13). After specifying user-named notepads, you are prompted for the name of the notepad document (and the folder in which it resides) each time a notepad function is requested.

Help Text Documents

Help text documents are documents created using OfficeVision/400 that contain help text for end-user applications. These documents contain special help tags that mark the location of help text that is specific to a particular field or area of your application displays. Help text documents are created by entering help text label instructions prior to each individual section of help text, and then saving the resolved output of that document when it is printed. The document you specify to contain the resolved output is the help text document.

Queries

Queries on the AS/400 system are unique objects of type *QRYDFN. These objects reside in user libraries. Queries are created using the AS/400 Query product. Although queries often are used by the OfficeVision/400 word processing program when merging data and text to produce group mailings or customer reports, you cannot create queries using OfficeVision/400.

Queries contain data from a database file based on the selection specified in the query definition. For example, a query can be created from a very large customer file by selecting only those customers who owe more than \$500. Queries can access data from a single file, or from several different database files.

A query can be specified as the source of data for an AS/400 data/text merge function on the Merge Data Options display. This display is available from the following word processing displays:

- Data Field Instruction
- Data Field Heading Instruction
- Conditional Text Instruction
- Print Options (PRTDOC command)
- Merge Options (MRGDOC command)

The Work with Queries display can be accessed directly from the OfficeVision/400 Edit display by selecting option 1 (Query) from the Functions display (F17).

Object Naming Conventions

This section describes naming conventions for text profiles, dictionaries, database files, notepads, help text documents, and queries.

The following naming conventions apply to text profiles:

- The system profile is named SYSTEM
- All other text profiles are user-named
- All text profiles reside in folder QPRFFLR

The following naming conventions apply to dictionaries:

- IBM-supplied language dictionaries reside in library QDCT. See the text profile naming conventions for individual language dictionary names.
- User-defined dictionaries are user-named.

Database files used by OfficeVision/400 word processing are user-named.

The following naming conventions apply to notepads:

- System-named notepads:
 - Folder = QWPDOCS
 - Notepad = User ID
- User-named notepads are named by the user and can reside in any folder.

Help text documents are user-named and can reside in any folder.

Queries are user-named and can reside in any user library.

Word Processing Services Command Interfaces

This section describes the application interfaces of the word processing services. The description includes the control language (CL) interface.

Control Language Commands

Several CL commands are supported that allow the user to easily access a wide variety of word processing functions (see “Word Processing Services Control Language Interfaces” on page 8-14).

CL commands can be called in one of the following ways:

- From the command line of any AS/400 menu. This allows the user to bypass one or more OfficeVision/400 menus, thereby reducing the number of steps users must perform.
- From a user-defined menu. When defining a menu using SDA (screen design aid), the user can specify a command or program to be called for each option on the menu.
- From a CL program. The CL program can contain several CL commands that are processed in the desired order. Therefore, calling one CL program can automatically perform several functions otherwise performed individually. CL commands are entered into a source file using the EDTSRC command. CL programs are created using the CRTCLPGM command.

Other Word Processing Services Topics

This section describes other word processing service topics such as editor selection, document interchange, text recovery, designing online help with help text, and security considerations.

Editor Selection

When using PC Support/400, the CHGPCOPRF command shows the display where you can specify which editor to use for word processing functions. The choices are DisplayWrite 4 (DW4), DW4 Version 2, and OfficeVision/400. If OfficeVision/400 word processing is selected, the PC text-assist version of the AS/400 word processing editor is used.

The CHGPCOPRF command performs the same function as selecting option 6 (Organizer profile options) from the PC Support/400 organizer menu.

Choosing between AS/400 Word Processing Editors

The Change Enrollment Information display can be used to choose a text editor. When enrolling a new OfficeVision/400 user, or changing a user's enrollment record, the system administrator can specify which version of the AS/400 word processing editor should be used.

The word processing option can be selected from the Change Enrollment Information display. It has two possible choices for word processing:

1. Standard or PC text-assist
2. Adapted

Selecting option 1 causes one of the text-assisted versions of the editor to be used, depending on whether PC Support/400 is being used. Option 1 will use adapted word processing if you are using a work station that does not support either of the text-assisted versions.

Selecting option 2 causes the adapted word processing editor to be used no matter what type of work station is being used. This version of the editor can perform word processing functions when the work station is not attached to a controller with the text-assist feature. You can perform nearly all the functions of the normal word processing editor using the adapted editor. Refer to the *Using OfficeVision/400* Adapted Word Processing Function* manual for a complete list of the differences.

Text Recovery

When a document is being changed, it is actually a copy of the document being processed. If the changes somehow end abnormally, as when an internal error occurs, the system keeps both the old and the current copies of the document so the user has the option of recovering the changes the next time the document is changed. A document is in a recovery state if the above is true.

The CHKDOC and PAGDOC commands cause the document to be updated. Therefore, these commands cannot be successfully completed using a document that is in a recovery state. If you attempt to use these commands on a document in a recovery state, the system sends message OFC8011. Watch for this message in all CL programs using the CHKDOC and PAGDOC commands.

The EDTDOC command allows you to recover a document that is in a recovery state. When EDTDOC or WRKDOC (and revise option) is used, the system displays the Recover Document display from which you can choose to recover the changes made to the interrupted document or to delete the changes and return to the interrupted document. If you choose to recover the changes, you still have the option not to save these changes when ending the edit session.

The remaining word processing service commands do not cause the document to be updated. Therefore, these commands can be successfully completed using a document that is in a recovery state. However, the commands use the old copy of the document, which may not be current.

Document Interchange

This section offers some considerations for interchanging (sending) documents or notes to users on other systems.

When documents or notes are sent from one system to another, the document or note must be converted into a format recognized by both of the systems. For documents, this format is called revisable-form text document content architecture (RFTDCA). For notes, this format is called final-form text document content architecture (FFTDCA). It is not always possible to convert all the data from the various internal formats, such as RFTAS400, to RFTDCA or FFTDCA.

The following document constructs will not interchange when a document or note is sent from an AS/400 system to any other system, including another AS/400 system, or from a System/36 to an AS/400 system:

- Help label parameters of index and outline heading text instruction
- Print options
- Text month names
- Numeric editing (math format)
- Spelling options

The sections that follow deal with specific cases of interchanging documents between systems.

AS/400 System to AS/400 System

If the receiving AS/400 system is at an OS/400 release level lower than the sending AS/400 system, some of the document data may not be recognized. For example, when a document is sent from a Release 2.0 AS/400 system to a Release 1.2 AS/400 system, the set variable text instruction will not be recognized if the document is edited, printed, or merged on the Release 1.2 AS/400 system.

AS/400 System to System/36

Sending a document from the AS/400 system to the System/36 is similar to sending between AS/400 systems, except there are several more document constructs that will not be recognized. (Footnotes are one example.)

AS/400 System to Other Systems

In general, the amount of data that could be lost when sending from an AS/400 system to other systems is dependent on the level of RFTDCA supported by both systems. Refer to the *Revisable-Form Text Reference* manual for more exact information about interchanging between AS/400 systems and other systems.

Designing Online Help with Help Text Documents

Help text documents allow you to design online help. There are three major steps to designing online help with help text documents:

1. Create the help text document.
2. Define help areas for the specific application display file.
3. Access the help text.

Each of these steps is described in detail in the following sections.

Creating the Help Text Document

Follow these steps to create a help text document:

1. Create a document to contain the help text source. This document is created in the same way as any other document.
2. Place a help text label instruction at the beginning of the document for general information about the application display. To do this, follow these steps:
 - a. Press F5.
 - b. Type .help.
 - c. Press the Enter key.
 - d. Type a label name.
 - e. Do NOT type a document and folder name unless referring to help text that is stored in another document.
3. Type general information about the display for which you are creating help text. This information appears when the Help key is pressed while the cursor is not on one of the help areas defined in the DDS for the display.
4. Perform the following for each individual prompt in the display for which you are writing the help text:
 - a. Insert a help text label instruction with a label name that describes the prompt (for example, CUSTNBR, NAME, ADDR, and STATE).
 - b. Type the help text for the prompt immediately following the help text label.
5. If desired, insert help text label instructions for the following special areas of the help text document:

- Insert a help text label called TOC immediately before the table of contents instruction (.toc) if your help text document contains a table of contents.

Note: The Outline Heading Instruction (.h) display contains two options specifically for help text documents. The *Start on new display* and *Show heading text* prompts give you greater control over where and how your help text information appears in the resolved help text document.
 - Insert a help text label called INDEX immediately before the index instruction (.ix) if your help text document contains an index.

Note: If you specify Y for the *Create help tags* prompt on the Index Instruction display, help text labels are automatically created for each letter section of the index when your document is resolved. This allows you to type, for example, the letter s to go directly to the index words starting with s when viewing the help document.
 - If your help text document contains a glossary, insert a help text label called GLOSSARY immediately before that section.
6. When you finish creating your help document source, press F3 (Exit), type Y for the *Print document* and *Display print options* prompts, and press the Enter key.
 7. Page forward to page 3 of the Print Options display.
 8. Specify Y for the *Save resolved output* prompt, specify a unique document name for the *Document* prompt, and also specify the folder where you want the help text document stored.
 9. This resolved document is your help text document. This is the document name that is specified in the data description specification (DDS), for the application display.

Defining Help Areas for Your Application Display File

Your application display files must contain references to the same help text labels and the help text document, where they are located. Once the help text labels are defined in both the display files and the help text document, simply press the Help key to take your users directly to the appropriate section of the help text document.

Help text labels can be specified in your display files in one of two ways:

1. Coding the DDS specifications directly
 - At the file level:
 - HLPDOC Keyword (Help document).
Format: HLPDOC(label document folder).
Example: HLPDOC(GENERAL HLPDOC HELPFLR).
The label specified here should show general information about the display.

– HELP Keyword (Help key). This activates the HELP key for the display file.

• At the record level:

– HLPDOC Keyword (Help document).

Example: HLPDOC(STATE HLPDOC HLPFLR).

– HLPARA Keyword (Help area).

Format: HLPARA(beg. row beg. col end row end col).

Example: HLPARA(11 1 11 80).

The HLPDOC and HLPARA keywords are coded in pairs, with the HLPARA keyword immediately following the corresponding HLPDOC keyword. One pair is coded for each help area on your display file.

2. Using screen design aid (SDA), select the Design Screens option and respond to the following prompts:

• File definition

a. Design option = 1 (Select file keywords).

b. Specify file name.

c. Select File Keywords display appears. Specify Y for:

– Indicator keywords

– Help keywords

d. Define Indicator Keywords display appears. Specify HELP as an unconditional keyword.

e. Select Help Keywords display appears.

1) Specify the label for general display file help text. This help text appears if the cursor is NOT within a defined help area when HELP is pressed.

2) Respond to the following prompts as shown below:

Help text in document Y

Record or document Help text document name

HLPDOC label Label name

HLPDOC folder Folder where help text document is stored

• Record definitions

a. Design option = 2 (Select record keywords).

b. Specify record name and file name.

c. Select Record Keywords display appears. Specify Y for application help.

d. Define Application Help display appears.

1) Specify the label for each prompt on the application display. Must also specify the help area for each prompt.

2) For each prompt specify:

Next help number Relative number (1-n)

Help text in document Y

| | |
|---------------------------|---------------------------------|
| <i>Record or document</i> | Help text document name |
| <i>HLPDOC label</i> | Label name for this prompt |
| <i>HLPDOC folder</i> | Folder where document is stored |

- e. The Define Help Area display appears. Specify row and column boundaries for each possible prompt position, or specify Y for the *Entire record is one area* prompt. The following example shows how to specify row and column boundaries:

```
Row/column boundaries .... From (row) (column)
                          To   (row) (column)
```

If the prompt for ADDRESS is on line 10 of the display file, and you want to show help text for ADDRESS when the cursor is anywhere on this line as the Help key is pressed, specify:

```
From 10 1
To   10 80
```

- f. Exit and save changes, then create the display file.

Accessing the Help Text

Access the help text using one of the following methods:

- Access the help text from an application display with the Help key by doing the following:
 1. Position the cursor at the prompt for which you need help.
 2. Press the Help key.
 3. The help document is opened and your cursor is positioned at the information in the document that describes the prompt where your cursor was located when you pressed the Help key.
- The DSPHLPDOC command allows you to display the help text document without running an application. This command can be entered in one of the following ways:
 - Specify the document name and folder in the DSPHLPDOC command to display the beginning of the help text document.
 - Specify a document name, folder, and label with the DSPHLPDOC command to display the help text associated with the specified label.

Functions for Viewing a Help Text Document

The following functions are available while viewing a help text document:

- Page forward or backward to view other information in the help text document.
- F9 = Set bookmark and exit.
 - Allows you to return to the same location in the help text document the next time you press the Help key.
 - Supported only for the Help key. Not supported for the DSPHLPDOC command.
- F14 = Print the help text document.

- A special *Go to* prompt allows you to quickly move to other areas of the help text document:
 - Go to another page by typing a page number (example: 5)
 - Go to another section by typing a section number (example: 5.2)
 - Go to the beginning of the index by typing INDEX
 - Go to any letter section of the index by typing the letter (example: S)
 - Go to the table of contents by typing TOC
 - Go to the glossary by typing GLOSSARY

Using the Run Instruction

The Run instruction is a text instruction that allows a CL command to be run when an AS/400 document is printed or when sending a note in OfficeVision/400. If the CL command produces printed output, the output appears in the printed output at the location where the run instruction was inserted. Commands that do not create printed output, but instead start some type of batch processing, can also be performed with the Run instruction.

The use of the Run instruction is limited based on enrollment information. Under the authority section of the Change Enrollment Information display, you must specify Y = yes for the *Allow commands in documents* prompt. If you are not allowed to use commands in documents, Run instructions are ignored when a document is printed.

To insert a Run instruction into a document:

1. Create or change a document.
2. Do one of the following:
 - a. Press F9 to show the Text Instructions menu, then select option 26 (Run instruction) and press the Enter key.
 - b. Press F5 (Goto), then type `.run` and press the Enter key.

The Run Instruction display appears.

3. Type the CL command you want to run. The F4 prompt is supported to display lists of commands or to prompt for a specified command. You can also enter System/38 CL commands, but you must specify 'Y' for the System/38 command prompt. For instance, you could specify the CALL command to call a user application that creates a printed report. The printed output would then be included with the document output when the document is printed.
4. Specify the print file to be used for the printed output. If you are running an IBM-supplied CL command, you must use the name of the print file found in the *CL Reference* manual. If you are using a user-written CL command, you must use the name of the job file created by the CL command or the print file used by the CL command. If nothing is specified for this prompt, QSYSPRT is used.
5. Press the Enter key to insert the Run instruction into your document at the cursor location.

Security Considerations

If you are printing a document that contains a Run instruction, you must have at least *USE authority to the command being run and any command or programs that this command calls. In addition, you must have at least *USE authority to the following CL commands:

- CRTPF (Create Physical File)
- OVRPRTF (Override Printer File)
- DLTSPLF (Delete Spooled File)
- DLTOVR (Delete Override)

Word Processing Services Control Language Interfaces

This section describes the following control language interfaces:

- Check Document (CHKDOC) command
- Copy Document (CPYDOC) command
- Create Document (CRTDOC) command
- Create Spelling Aid Dictionary (CRTSPADCT) command
- Delete Spelling Aid Dictionary (DLTSPADCT) command
- Display Document (DSPDOC) command
- Display Help Document (DSPHLPDOC) command
- Edit Document (EDTDOC) command
- Fill Form Document (FILLFORM) command
- Merge Document (MRGDOC) command
- Paginate Document (PAGDOC) command
- Print Document (PRTDOC) command
- Print Stop Word List (PRTSWL) command
- Retrieve Stop Word List Source (RTVSWLSRC) command
- Start Word Processing (STRWP) command
- Work with Documents (WRKDOC) command
- Work with Document Print Queue (WRKDOCPRTQ) command
- Work with Folders (WRKFLR) command
- Work with Text Profiles (WRKTXTPRF) command

Check Document (CHKDOC) Command

The Check Document (CHKDOC) command verifies the spelling of a document or performs a grade level analysis on a document. It allows you to spell check a document while editing another. The spelling function completes and you receive a message when the checking is completed. For additional detailed information on the CHKDOC command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

The user must have at least *USE authority for the dictionaries to be used by the CHKDOC command. The user must have authority to update the document specified in the CHKDOC command. This command is shipped with *USE authority, which means all users can use the command.

Some Ways to Use This Command

You can check the spelling in a large document without affecting the performance of others on the system. CHKDOC allows you to submit a spell check and perform other office functions while it is running.

Examples:

```
CHKDOC DOC(MYDOC) FLR(MYFLR) TYPE(*GRADE) STRPAGE(1) ENDPAGE(50)
CHKDOC DOC(MYDOC) FLR(MYFLR) TYPE(*SPELL) STRPAGE(1) ENDPAGE(50)
```

These two examples check the document MYDOC in folder MYFLR starting on page 1 and ending on page 50. The first performs a grade level analysis while the second performs a spelling verification.

```
CHKDOC DOC(MYDOC)
```

This checks the spelling of the document MYDOC. It looks for this document in the last folder the user accessed. The entire document is checked for spelling errors.

Rules

You must have OfficeVision/400 loaded on the system in order to use this command. The document specified must have a dictionary in its dictionary search list that supports spell check, if *SPELL was specified. Furthermore, it must support grade level checking, if *GRADE was specified. The dictionary search list is specified on the Dictionary Options display.

A document specified for a *GRADE type of check must have a grade level associated with its text. This is done on the Spell Options display. Grades 4-16 are supported. Only the U.S. (American English) dictionary supports grade level checking.

Once the spell check is complete, the user must change the document and search for misspelled words. This can be done by using the Locate marks option on the Spell Options display.

Hints and Techniques

The document text is checked using the code page of the document.

Parameters

The CHKDOC command uses the following parameters:

DOC Document to check.

FLR Folder where document resides.

OPTION Type of checking to perform (*SPELL or *GRADE).

STRPAGE Page to start checking (*BEGIN, *FIRST, or page number). *BEGIN or *FIRST specifies the beginning of the document.

ENDPAGE Page to end checking (*END, *LAST, or page number). *END or *LAST specifies the end of the document.

Create Document (CRTDOC) Command

The Create Document (CRTDOC) command creates a new document in a specified folder. Using this command allows you to bypass the OfficeVision/400 menu, the Word processing menu, and the Work with Documents in Folders display, and takes you directly to the Create Document Details display or the word processing Edit display.

Security Considerations

This command is shipped with *USE authority, which means all users can use this command. Use the GTROBJAUT command to restrict the use of CRTDOC for specific users.

```
GRTOBJAUT OBJ(QSYS/CRTDOC) OBJTYPE(*CMD) USER(ABC) AUT(*EXCLUDE)
```

A user must have at least *CHANGE authority for a folder to be able to create a document in that folder.

Some Ways to Use This Command

The CRTDOC command can be entered from a command line or called from a CL program.

The following Create Document command creates a document named REPORT in a first-level folder named INTERNAL.

```
CRTDOC DOC(REPORT) FLR(INTERNAL)
```

The following Create Document command creates a document named MEETING in a second-level folder named DEPT800, which resides in the folder DEPTS.

```
CRTDOC DOC(MEETING) FLR('DEPTS/DEPT800')
```

The following Create Document command creates a document named SALES in the folder you worked with previously.

```
CRTDOC DOC(SALES)
```

Rules

The document created by the CRTDOC command is made by copying the currently active text profile document. The default text profile is SYSTEM, in the profile folder QPRFFLR. The system profile is not used if a document to copy is specified on the Create Document Details display, or if a folder profile or user profile is active.

The create function first checks to see if a document was specified for the *Document to copy* prompt on the Create Document Details display. If so, this document is copied to create the new document.

The create function next checks to see if a profile was assigned to the folder in which the document is being created. If a profile was assigned, it is copied to create the new document.

If a profile has not been assigned to the folder, the create function then checks to see if a text profile was activated by the user. If a text profile was activated, it is copied to create the new document.

If a text profile is not active for either the folder or the user, then the system profile (SYSTEM) is copied. Text profiles can be created or activated using the WRKTXTPRF command. Refer to “Work with Text Profiles (WRKTXTPRF) Command” on page 8-43 for more information about creating and activating files.

When creating documents, limit the number of apostrophes in the name to two. The MRGDOC command cannot process document names with more than two apostrophes if submitted in batch.

Hints and Techniques

The Create Document command shows the Create Document Details display when the DETAILS parameter has a value of *YES. You must supply the information on this display interactively. If you need to create a new document in a batch environment, you can use the Copy Document (CPYDOC) command to copy an existing document.

Character Set and Code Page Considerations

The document created by the CRTDOC command assumes the character set and code page of the active text profile document, or the document specified by the *Document to copy* prompt on the Create Document Details display. The active text profile is displayed at the top of the Create Document Details display.

Parameters

Following are the CRTDOC parameters:

- | | |
|----------------|--|
| DOC | Document to create. |
| FLR | Folder where document will reside. |
| TXTPRF | Text profile to use for this create request. |
| TEXT | Text description of this document. |
| DETAILS | Display the Create Document Details display. Possible values are: *YES *NO |
| EDIT | Edit the text of the document to be created. Possible values are: *YES *NO |
| EXITPNL | Display the Exit Document display. Possible values are: *YES *NO |

Create Spelling Aid Dictionary (CRTSPADCT) Command

The Create Spelling Aid Dictionary (CRTSPADCT) command allows you to create a user-defined spelling aid dictionary or stop word list. The input for this command is a member of a source file containing the words to be included in the dictionary or stop word list.

Stop word lists are a special type of spelling aid dictionary used by text search services during linguistic processing of documents. Text search services uses stop word lists to store common words that are not used in searches. Stop word lists can be either supplied by IBM or created by the user using the CRTSPADCT command. User-created stop word lists are stored in the QUSRSYS library with

names of QTWnnnX (where *nnn* is the language ID) and a type of *SPADCT. See "Stop Word Lists" on page 7-13 for an example of how to create a stop word list using the CRTSPADCT command.

Security Considerations

The user must have *USE level authority to the member, file, and library being used as input for this command. The created dictionary will have *CHANGE level authority unless specified otherwise by the user in the AUT parameter. This command is shipped with *USE authority, which means that all users can use the command. The user must have *USE authority for the Clear Physical File Member (CLRPFM) command and the Create Physical File (CRTPF) command.

Some Ways to Use This Command

Sometimes the OfficeVision/400 editor highlights a word as misspelled even though it is spelled correctly. This happens because the word was not found in any of the dictionaries in your dictionary search list. Such a word can be a proper name, an abbreviation, or a newly coined word. You can create a user-defined dictionary to contain the words you consider valid, but which are not in any of the IBM-supplied language dictionaries. In fact, you can create more than one user-defined dictionary (for example, a dictionary for abbreviations and another dictionary for proper names). When you add your user-defined dictionary to your dictionary options search list, the words in the dictionaries are considered valid.

Following is the command entered with just the required parameters:

```
CRTSPADCT SPADCT(MYDCT) SRCFILE(MYSRC)
```

Following is the most commonly used set of parameters for the command:

```
CRTSPADCT SPADCT(MYLIB/MYDCT) SRCFILE(MYLIB/MYSRC)
SRCMBR(MYMBR) OPTION(*NOSRC) LNGATR(*ENGLISH)
REPLACE(*YES) TEXT('User dictionary containing company
abbreviations and proper names')
```

The following command creates a dictionary called MYDCT in library MYLIB. The input words come from the source file member MYMBR, which is in the file MYSRC in library MYLIB.

```
CRTSPADCT SPADCT(MYLIB/MYDCT) SRCFILE(MYLIB/MYSRC) SRCMBR(MYMBR)
OPTION(*SRC) BASEDCT(MYLIB/MYBASDCT) VFYDCT(QDCT/US)
LNGATR(*ENGLISH) AUT(*CHANGE) REPLACE(*YES) TEXT('User
dictionary containing company abbreviations and proper names')
```

Any words found in this source file member that also exist in the U.S. dictionary are not added to the new dictionary. All words in the base dictionary MYBASDCT are added to the new dictionary. The dictionary has the language attribute of ENGLISH, which means the words entered in the dictionary follow the English grammar rules when this dictionary is used within a document for spell functions. If the dictionary MYDCT already exists in library MYLIB, it is deleted before the new dictionary is created. A printout of the job log from this command is created by the *SRC option.

Hints and Techniques

The CRTSPADCT command can be run on the job queue. When creating large dictionaries you may want to consider using the job queue.

Character Set and Code Page Considerations

The first 3 characters of the first line in your input source file member are always checked for a valid keyboard ID. If you do not specify a keyboard ID, the system value for the character set and code page is used. Or you can specify a CHRID in the last record of your source file member. A CHRID is a combination of a graphic character set identifier and a code page identifier. The format of the CHRID is:

```
CHRID  aaaaabbbbb
```

In the example, aaaaa is the graphic character set identifier and bbbbb is the code page identifier.

Parameters

Following are the CRTSPADCT parameters:

- SPADCT** Name and library of the spelling aid dictionary being created. For example:
- ```
SPADCT(Library-name/Dictionary-name)
```
- Use the special value \*USRSWL to create a user-defined stop word list in the QUSRSYS library with an IBM-supplied name.
- SRCFILE** Name and library of the source file containing the source member that contains the words for this dictionary. For example:
- ```
SRCFILE(Library-name/Filename)
```
- SRCMBR** Name of the source member that contains the words for this dictionary.
- OPTION** Source list option. Specifies the type of output list produced when the dictionary is created.
- *SOURCE or *SRC**
List the source statements
- *NOSOURCE or *NOSRC**
Do not list the source statements
- BASEDCT** The name and library of the dictionary containing words to be added to the dictionary being created. A spelling dictionary (IBM-supplied) cannot be specified here. For example:
- ```
BASEDCT(Library-name/Dictionary-name)
```
- Note:** This parameter is not allowed when SPADCT(\*USRSWL) is specified.
- VFYDCT** The name and library of an existing dictionary that is searched for each word specified in the source member. Only those words NOT found in the verify dictionary are placed in the new dictionary to avoid duplication. For example:
- ```
VFYDCT(Library-name/Dictionary-name)
```
- Note:** This parameter is not allowed when SPADCT(*USRSWL) is specified.

LNGATR The language attribute to be associated with the dictionary being created. The language attribute determines the processing rules that apply when the dictionary is used.

***VFYDCT** Use same language attribute as the dictionary specified as the verify dictionary.

***NONE** No specific language attribute.

Specific language attributes include the following:

| | | |
|-----------|-----------|-----------|
| *AFRIKAAN | *ESPANA | *NORSK |
| *BRASIL | *FRANCAIS | *PORTUGAL |
| *CATALA | *FRA2 | *SUOMI |
| *ENGLISH | *GREEK | *SVENSK |
| *DANSK | *ISLENSK | *TURKISH |
| *DEUTSCH | *ITALIANO | |
| *DSCHWEIZ | *NEDERLND | |

Note: This parameter is not allowed when SPADCT(*USRSWL) is specified.

AUT The authority for the spelling aid dictionary to be granted to other users who do not have specific authority to the dictionary. Possible values are:

- *CHANGE
- *ALL
- *USE
- *EXCLUDE

REPLACE Replace an existing spelling dictionary with the same name and library. Possible values are:

- *YES
- *NO

TEXT Text description of this spelling aid dictionary.

SWLLANGID The language for the stop word list. This parameter is allowed only when SPADCT(*USRSWL) is specified.

IBMSWL Include the IBM-supplied stop word list in the user-created stop word list. Possible values are:

- *YES
- *NO

This parameter is allowed only when SPADCT(*USRSWL) is specified.

Delete Spelling Aid Dictionary (DLTSPADCT) Command

The Delete Spelling Aid Dictionary (DLTSPADCT) command deletes the specified spelling aid dictionary from the system.

Security Considerations

You must have *ALL level authority for the spelling aid dictionary being deleted. This command is shipped with *USE authority, which means all users can use the command.

Some Ways to Use This Command

The following command deletes the dictionary MYDCT from library MYLIB:

```
DLTSPADCT SPADCT(MYLIB/MYDCT)
```

The following command deletes all dictionaries that begin with the characters MY in libraries in your current library list:

```
DLTSPADCT SPADCT(MY*)
```

Hints and Techniques

Deleting the dictionary does not cause the originating source file and member to be deleted. You must use the RMVM command to remove the associated source file member from its file if you no longer wish to use that source file member as input or you can use STRSEU interactively to delete members.

DLTSPADCT uses the SPADCT parameter. As the following example shows, it specifies the name and library of the spelling aid dictionary being deleted.

```
SPADCT(LIBNAME/DICTNAME)
```

You also use the DLTSPADCT command to delete user-created stop word lists. The lists are in library QUSRSYS with the name QTW nnn X, where nnn is the language ID of the list.

Display Document (DSPDOC) Command

The Display Document (DSPDOC) command displays a document within a specific folder. This command performs the same function as option 5 (View) on the Work with Documents in Folders display. You cannot make changes to a document using the DSPDOC command.

Security Considerations

The user must have at least *USE level authority to display the document. This command is shipped with *USE authority, which means that all users can use the command.

Some Ways to Use This Command

The DSPDOC command can be called from a CL program to allow you to view a document that should not be changed. For instance, selecting an option from an application menu that calls the DSPDOC command allows you to view an example of a particular type of documentation. This is beneficial to new users working in an office where strict documentation standards are followed.

This command can also be called from a CL program to view help text for the application. By using the DSPDOC command, the help text document does not have to be resolved (as it does when using the DSPHLPDOC command).

The following Display Document command displays the document SHELL.FRM, which resides in the folder SHELLS:

```
DSPDOC DOC(SHELL.FRM) FLR(SHELLS)
```

The following Display Document command displays the document SHELL.LET, which also resides in the folder SHELLS. Since SHELLS was the folder last used by this user, the folder does not have to be specified:

```
DSPDOC DOC(SHELL.LET)
```

The following Display Document command displays the last document used by this user. Since the document SHELL.LET in folder SHELLS was the last document used by this user, that document is displayed:

```
DSPDOC
```

The following Display Document command displays the document SHELL.LET, which also resides in the folder SHELLS. The user cannot print the document while viewing it:

```
DSPDOC DOC(SHELL.LET) ALWPRT(*NO)
```

Rules

The DSPDOC command can be used to display all the types of documents that OfficeVision/400 Word Processing can work with, unlike the DSPHLPDOC command, which can only display resolved documents.

Parameters

Following are the DSPDOC parameters:

| | |
|---------------|--|
| DOC | Document to display. |
| FLR | Folder where the document resides. |
| ALWPRT | Specifies whether the user is able to print a document while viewing it. Possible values are: *YES *NO |

Display Help Document (DSPHLPDOC) Command

The Display Help Document (DSPHLPDOC) command displays help text information stored in a help text document without having to run an application and press the Help key. A help text document is created by resolving a document that contains Help Text Label instructions (.help). Such documents can be accessed from a user application display by pressing the Help key, or by specifying the help text document, folder, and label names in the definition of the display file.

Security Considerations

The user must have at least *USE level authority for the help document to display it. This command is shipped with *USE authority, which means all users can use the command.

Some Ways to Use This Command

The DSPHLPDOC command displays help text in the format it would appear if you had pressed the Help key from an application that accesses help text stored in a document. The user can view help text without having to start an application and press the Help key on a specific display. After creating a help text document, the writer can use this command to examine the help text prior to linking it to a heavily used application.

This command can also be called from a CL program to display directions for performing a given task, or to display a description of an application. Selecting an option on an application display can call this command and cause help text to appear.

The following command displays the help document HELP100 that resides in folder HELPTXT:

```
DSPHLPDOC DOC(HELP100) FLR(HELPTEXT)
```

The beginning of the help document HELP100 is shown, since no specific help label has been specified.

This Display Help Document command displays the help document HELP200 that resides in folder HELPTEXT. The section of the help document immediately following the help text label STATE is shown initially:

```
DSPHLPDOC DOC(HELP200) FLR(HELPTEXT) HLPBLB(STATE)
```

The label name must have been specified for a help text label instruction that was inserted into the help document prior to resolving it.

Rules

A help document cannot be displayed using the DSPHLPDOC command, or by pressing the Help key on an application display, unless the document is a resolved document. A help text document is resolved by specifying Y=Yes for the *Save printed output* prompt on the third page of the Print Options display, then specifying a document and folder name to store the output of the help document. You may also resolve a help text document when using the PRTDOC command by specifying the SAVOUTPUT, SAVDOC, and SAVFLR keywords. It is the document and folder name you specify on the third page of the Print Options display that you specify for the DSPHLPDOC command, or when specifying your application's help text when using DDS or SDA.

When displaying a help text document, a special *go to* prompt appears at the bottom of the text, allowing you to type a section number, page number, or the name of a special part of the help text that you want to see. If you type a number and do not follow it with a period (2), you will go to that page number. If you type a number and follow it with a period (2.), you will go to that section number. Section numbers are created in a help text document by inserting outline heading instructions into the help text source document. When the source document is resolved, the outline heading instructions are turned into section numbers.

The user can also enter the special values TOC, GLOSSARY, or INDEX to go directly to those areas. TOC takes you to the table of contents, which is created by inserting a table of contents instruction (.toc) into the help text source document. GLOSSARY takes you to the glossary section, which must be preceded by a help text label with the name GLOSSARY.

INDEX takes you to the beginning of the help document's index, which is created by inserting an index instruction (.ix) into the help text source document. The user can also enter a single letter to go to the section of the index that shows entries that begin with that letter. However, index entries are not automatically created in a help text document. The user must enter index entry instructions (.ie) into the help text source document for each entry to be shown in the index.

Parameters

Following are the DSPHLPDOC parameters:

- DOC** Help document to display.
- FLR** Folder where help document resides.

HLPLBL Specifies where to start displaying the help document. If a specific label is not specified, the beginning of the help document is displayed.

Edit Document (EDTDOC) Command

The Edit Document (EDTDOC) command allows the user to make changes to a document within a specific folder. This command performs the same function as option 2 (Revise) on the Work with Documents in Folders display. Using the EDTDOC command, however, allows you to bypass the OfficeVision/400 menu, the Word Processing menu, and the Work with Documents in Folders display and go directly to the word processing Edit display.

Security Considerations

The user must have at least *CHANGE level authority for the document to edit it. The user does not have to be enrolled in OfficeVision/400 to use this command. For instance, if the user is using PC Support/400 organizer and has specified DW4 as the text editor of choice using CHGPCOPRF, the EDTDOC command can be used to edit a PC document with DW4. This command is shipped with *USE authority, which means that all users can use the command.

Some Ways to Use This Command

Many offices have standard form letters that they use several times a day for customer correspondence. A menu could be created for the users who fill in these forms, with a separate option for each type of form. Selecting an option from this menu would first run the CPYDOC command to copy a standard shell document to the work document, replacing the existing work document. The EDTDOC command would then be run for the work document to take the user directly to the Edit display to fill in the form. By using these commands, the user has bypassed performing two separate functions from the Work with Documents in Folders display (3=copy, 2=revise) by simply selecting an option from a menu.

The EDTDOC command can be called in a CL program, immediately after a Merge Document (MRGDOC) command is called, to edit the resulting document from the merge function. This allows the user to edit a document that contains data merged from a file, query, or fill-in document.

The following Edit Document command edits the document SHELL.FRM, which resides in the folder SHELLS:

```
EDTDOC DOC(SHELL.FRM) FLR(SHELLS)
```

The following Edit Document command edits the document SHELL.LET, which also resides in the folder SHELLS. Since SHELLS was the folder last used by this user, the folder does not have to be specified.

```
EDTDOC DOC(SHELL.LET)
```

This Edit Document command edits the last document used by this user.

```
EDTDOC
```

Since the document SHELL.LET in folder SHELLS was the last document used by this user, that document is changed.

Rules

Two or more users cannot edit the same document at the same time. If the same document needs to be edited by more than one user at any given time, the document can be copied into different folders.

The word processing editor can only edit a document that is in AS/400 text internal format. If the EDTDOC command is used for a document that is not already in this format, for instance, RFTDCA format, a transformation is automatically run by the EDTDOC command to transform the document into AS/400 text internal form. The EDTDOC command cannot be used for a document containing PC data unless DW4 is the text editor of choice.

Using the Edit Document command for a resolved document is possible, but not advised, since the document is already in final (printed) format. A resolved document has already been divided into pages, with headers and footers. Adding or deleting lines of text can cause problems since the document is not resolved again when it is printed.

The End a Document display is always shown when F3 is pressed to exit an edit session started with the EDTDOC command. This display allows the user to save or not save the changes made during the edit session. There is currently no way to bypass this display when exiting an edit session.

Character Set and Code Page Considerations

When editing a document, newly entered text is stored in the character set and code page associated with the document, no matter what the character set and code page are for the work station being used, as long as the work station, document character set, and code page use one of the supported code pages. See Appendix A, "Character Set and Code Page Considerations," for more information.

Parameters

Following are the EDTDOC parameters:

| | |
|----------------|--|
| DOC | Document to change. |
| FLR | Folder where the document resides. |
| EXITPNL | Display the Exit Document display. Possible values are: *YES *NO |

Fill Form Document (FILLFORM) Command

The Fill Form Document (FILLFORM) command allows you to complete a form or shell document by filling in the form fields defined in the document. Using this command allows you to bypass the OfficeVision/400 menu, the Word Processing menu, and the Work with Documents in Folders display. The command takes you to the Fill Form Document display if you press F4. The command will take you directly to the Fill In Form Edit display using the previously accessed document as the default document to be filled in. Once the data is entered, the user can print or send the form (with the data), then refresh the form and begin entering another set of data, without having to end the Fill in Form session. For additional information, such as parameters, on the FILLFORM command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

The user must have at least *USE authority for a form field document to be able to fill in the document. If the form field document will be saved to another document specified in the command parameters, at least *CHANGE authority is required to the saved document. If any of the form fields will be saved to a database file, at least *CHANGE authority is required to that database.

If any of the form fields were defined as requiring validity checking using a user-defined exit program and database file, the user must have at least *USE authority to the program and file.

Some Ways to Use This Command

The FILLFORM command can be entered from a command line or called from a CL program.

For example, the following command uses the document REPORT in the folder ACCOUNT as the shell document and saves the filled-in document in the document MONDAY in the folder WEEKLY. If the document MONDAY already exists in the folder WEEKLY, then the *YES in the REPLACE parameter indicates the existing data will be replaced by the current data.

```
FILLFORM CPYDOC(REPORT) CPYFLR(ACCOUNT) FORMDOC(MONDAY)
FORMFLR(WEEKLY) REPLACE(*YES)
```

The following command uses the document REPORT in the folder ACCOUNT as the shell document. Status lines will not be shown on the fill form display, the exit document prompts will not be shown when F3 or F12 is pressed to end the edit session, and refresh will not be allowed.

```
FILLFORM CPYDOC(REPORT) CPYFLR(ACCOUNT) DSPSTSLNE(*NO)
EXITPNL(*NO) ALWRFSH(*NO)
```

The following command uses the document REPORT in the folder ACCOUNT as the shell document. The initial value 101 for the form field defined with the ID DEPT and the initial value MENS for the form field defined with the ID NAME will override the initial values stored in the shell document REPORT in the folder ACCOUNT.

```
FILLFORM CPYDOC(REPORT) CPYFLR(ACCOUNT)
INDATA((DEPT 101) (NAME MENS))
```

The following command uses the document REPORT in the folder ACCOUNT as the shell document. The data entered in the form fields defined with the IDs DEPT and NAME will be added to the database file member DECEMBER in the file named ACTSALES in the library named SALES when the form is refreshed, but not when the form is exited or if errors exist within the form data.

```
FILLFORM CPYDOC(REPORT) CPYFLR(ACCOUNT) OUTDATA((DEPT) (NAME))
OUTFILE(SALES/ACTSALES) OUTMBR(DECEMBER *ADD)
OUTOPT(*NO *YES *NO)
```

Rules

The FILLFORM command can be used only with documents that contain form fields. Form fields can be defined with the form field instruction when using any of the OfficeVision/400 word processing editors.

A person can use the FILLFORM command while the shell document is being edited in another session. Any additions or changes to the shell document will not be reflected in the form fill session until the edit session for the shell document is ended and the form fill session has ended, and is restarted.

Two or more users can use the FILLFORM command at the same time with the same shell document.

The FORM document (FORMDOC) cannot be the same as the COPY document (CPYDOC). In addition, if the FORMDOC already exists, the REPLACE (*YES) option must be specified. If the FORMDOC parameter is specified, then EXITPNL and ALWRFSH cannot both be *YES.

The user can only print the form field document interactively during a form fill session.

Hints and Techniques

If the user specifies data to be written to an output database file, that data can be used as input to other applications. The user can write a logical file that references the physical file where the data was written.

Character Set and Code Page Consideration

There is no translation between the command and the document. The assumption is made that the code page that is being used when the command is entered is the same as the code page used for the document.

Merge Document (MRGDOC) Command

The Merge Document (MRGDOC) command merges text and data, which allows the user to combine documents to create a new document that can be changed. For additional detailed information on the MRGDOC command, see Appendix D, "Control Language (CL) Commands." The MRGDOC command causes the following text instructions to be processed:

| | |
|--------------------|------------------------|
| *& | Data field |
| *dfh | Data field heading |
| *inc | Include |
| *bct | Begin conditional text |
| *ect | End conditional text |
| *set | Set |
| *&& | Variable |
| *run | *Run |

Security Considerations

The user must be enrolled in OfficeVision/400 to use this command. The user must have at least *USE authority for the document being processed and any data (file, query, document) merged into this document. The user must have at least *CHANGE authority to the folder that contains the new document being created. This command is shipped with *USE authority.

Some Ways to Use This Command

The most likely use of this command is with a data entry application. The data is entered into a file or document using DFU or some other data entry program. The MRGDOC command causes this data to be combined with an AS/400 shell document to create an editable report. An operator can then make minor changes to the resulting document, and use PRTDOC to actually print output.

For example, suppose there are five operators who answer telephone requests for information concerning a line of products that a company sells. When a customer calls, the operator asks a series of questions and, based on the answers, creates a letter that contains a set of paragraphs with the requested information. One way to create such an application is detailed in the following paragraphs.

Start with an AS/400 shell document to be used in a MRGDOC command. This shell document should contain any constant information for each letter (salutation, date by using a DATE instruction, and so on). This shell document should also contain Data Field text instructions to indicate the fields to include from the data entry application.

Use the MRGDOC command to merge a single record of data into a document. By specifying the merge type *RECORD, you can search a database for information using an application program and directly merge that information interactively into an object to be edited or viewed. You can specify up to 50 pairs of field name and data to merge combinations on the MRGDOC command.

Note: The parameter *RECORD supports up to 60 characters.

The application comes together as follows. As an operator receives a call, the necessary data is entered on the MRGDOC command. When the record is complete, the MRGDOC command for that operator is run. When the MRGDOC command is complete, the record is put into a master file. Use the PRTDOC command to print the merged output. The operator receives the printed customized letter for the person calling and is ready to take the next call.

The following command causes document MYDOC in folder MYFLR to be merged using member MYMBR of file MYDATA in library MYLIB as the merge data source:

```
MRGDOC FROMDOC(MYDOC) FROMFLR(MYFLR) TODOC(NEWDOC) TOFLR(*FRMFLR)
MRGTYPE(*FILE) DTAFILE(MYLIB/MYDATA) DTAMBR(MYMBR)
```

All Data Field text instructions found during the merge that have *PRINT as the merge source use MYMBR as the merge source. The merge creates a new document called NEWDOC in folder MYFLR.

The following command causes document MYDOC in folder MYFLR to be merged using the merge data source stored with the NEWDOC document:

```
MRGDOC FROMDOC(MYDOC) FROMFLR(MYFLR) TODOC(NEWDOC) TOFLR(*FRMFLR)
ADJUST(*BOTH) MLTLINRPT(*YES)
```

Both line and page endings of the resulting document NEWDOC are adjusted, and the NEWDOC document is formatted using multiple line report formatting.

If you specify *PRINT as the merge source, you need to specify the merge source. The merge source is remembered the next time you do run the MRGDOC command on this document only if you displayed merge options.

Rules

The FROM document (FROMDOC) cannot be the same as the TO document (TODOC). In addition, if the TODOC already exists, the REPLACE(*YES) option must be specified.

If you specify DSPOPT(*EDIT), DSPOPT(*VIEW), or DSPOPT(*FILLFORM), you cannot specify *YES on the Place on job queue option.

Hints and Techniques

The MLTLINRPT option can be used to create complex reports using a shell document that contains multicopy Data Field instructions. When this parameter is specified, the page ending control at the end of each copy of a multicopy document is removed.

During a merge using MRGDOC, special processing is done to look for INCLUDE text instructions that are embedded inside a field brought in by a Data Field instruction. Only the include instructions embedded inside a field specify *IMMED as one of the pages to include in the instruction. If a page number of *IMMED is used, the INCLUDE instruction is converted into an AS/400 system INCLUDE instruction and placed in the document (the INCLUDE is not performed until another MRGDOC or a PRTDOC is run). If a page number of *IMMED is not found, the INCLUDE instruction is converted into an AS/400 system INCLUDE instruction, and then the INCLUDE is performed (for example, the document referred to on the INCLUDE instruction is included). The function of *IMMED allows a different merge source to be specified on the data fields within the included document. These INCLUDE text instructions are useful in creating complex data entry applications that produce customized output.

Figure 8-1 shows how an INCLUDE embedded in a data field can appear (for example, Data Field2).

Figure 8-1. INCLUDE Embedded in a Data Field

| Data Field1 | Data Field2 |
|-------------|--------------------------------|
| Name1 | .inc(DOC1,FOLDER1,) |
| Name2 | .inc(DOC1,FOLDER1, 1 2 *IMMED) |
| Name3 | .inc(DOC2,FOLDER1, *IMMED) |
| : | : |

Note: The *IMMED option can be used only on an INCLUDE instruction in a file (for example, a page number of *IMMED to include cannot be used on an INCLUDE text instruction in a document). Only one *IMMED should be used per INCLUDE text instruction within a file.

A MRGDOC request can be placed on the job queue using the SBMJOB command or by specifying JOBQ(*YES) with the MRGDOC command.

You may want to vary the merge source when using this command. For example, you may have one standard shell and a different query for each region or period of time. If you expect to vary the merge source through the MRGTYPE keyword on the MRGDOC command, you must be sure to identify all Data Field and Begin Conditional Text text instructions with a merge source of *PRINT.

Paginate Document (PAGDOC) Command

The Paginate Document (PAGDOC) command causes the line endings and the page endings of an AS/400 document to be adjusted. The adjusting of lines is based on the amount of space between the left and right margins, and takes into account the zone width, the automatic hyphenation options, and paragraph boundaries. The adjusting of pages is based on the first and last typing lines, and takes into account keeps, running headings, table of contents, index, skips and skip to lines, graphics, footnotes, and multiple text columns. For additional detailed information on the PAGDOC command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

The user must be enrolled in OfficeVision/400 to use this command. The user also must have at least *CHANGE authority to the document being paginated. This command is shipped with *USE authority.

Some Ways to Use This Command

Since the pagination process can take time, you may decide to not use the interactive pagination functions available through the OfficeVision/400 editor. Instead, you can use the PAGDOC command to paginate all documents that were changed in a given day. An application program can be written to search through a folder looking for all documents with a last changed time of a given date. A PAGDOC command is started by that application for each document found. This application program can run off-shift to make optimum use of processing time.

The following command causes document MYDOC in folder MYFLR to be paginated:

```
PAGDOC DOC(MYDOC) FLR(MYFLR)
```

Rules

PAGDOC causes both line endings and page endings to be adjusted. There is no capability to request only line ending or only page ending adjustment.

PAGDOC always runs in the background.

Hints and Techniques

This command causes the source document to be paginated. Instructions like keeps, running headings, skips, and so on are taken into account during the pagination process. However, these instructions must remain a part of the source document, which may cause the document to display differently than it prints. For example, if a page has a skip instruction to skip 10 lines, this is taken into account during the page adjustment process, and this page will contain the proper amount of data after the PAGDOC command. However, since the skip instruction is not replaced with the corresponding amount of white space, the line count may not be accurate when viewed or edited.

Instructions like *includes* and *data* fields are not resolved during a PAGDOC, and therefore the space that may be created when these instructions are finally resolved is not accounted for during a PAGDOC. If you have documents that contain these kinds of instructions, use the option to paginate when a document is printed.

Print Document (PRTDOC) Command

The Print Document (PRTDOC) command allows you to print a document using the word processing function of OfficeVision/400. The Print Document (PRTDOC) command formats and prints a document that resides in an AS/400 folder.

The formatting process converts the document into final form. During the formatting process, the following is done:

- All text instructions are resolved. For example, a *DATE instruction becomes the system date, a *TOC instruction becomes a table of contents, data from documents referred to on all *INC (include) instructions is included, and data referred to on *& (Data Field) instructions is merged.
- Headers and footers are added to each page. If header or footer text exists, this text is also formatted.
- Line endings are optionally adjusted.
- Page endings are optionally adjusted (paginated).
- Line numbering and change flagging are optionally printed.

During the printing process, the final form document is converted into a format that can be processed by an output device. This output device could be a file, a printer, or a display. See the *CL Reference* manual for more detailed information on the PRTDOC command.

Security Considerations

The user does not have to be enrolled in OfficeVision/400 to use this command. The user must have at least *USE authority to the document being printed as well as other system objects (output queues, job descriptions, files, queries, and so on) that might be accessed during formatting or printing. This command is shipped with *USE authority.

Some Ways to Use This Command

A very common application is the creation of multiple letters by merging text and data. Typically, a shell document that contains some standard paragraphs is combined with variable data (name, address, and so on) from a file.

This variable data is entered into the file through an application program. Data Field text instructions in an AS/400 shell document mark the position where the data is to be inserted. The PRTDOC command can be used in the application program to merge the data created by the data entry application into the shell document, thus creating multiple printed letters.

Another useful application of this command is migrating data from systems that store files as text documents. Systems do exist that use their text editors as the data entry vehicle. Therefore, the data is actually nothing more than a document.

One method of migrating applications from these systems to the AS/400 system involves changing the application to use the AS/400 database. A necessary step in such a migration is to convert the data from a document into a database file. This can be done using the OPTIONS(*OUTFILE) function of the PRTDOC command (which creates a file containing records of a special, described format) with a query (using the special format of the above file) to create a new AS/400 database file.

See Appendix C, "Office Services Output File Considerations," for more information.

The following command causes document MYDOC in folder MYFLR to be formatted for printing on the printer you have specified in your user profile. The printed output goes to the default output queue for that printer:

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OUTPUT(*PRINT) DEV(*USRPRF)
OUTQ(*DEV)
```

The following command causes the document MYDOC in folder MYFLR to be formatted for printing to an output file:

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OUTPUT(*OUTFILE)
OUTFILE(MYLIB/MYFILE) OUTMBR(*FIRST) CURSTS(*NOCHK)
NEWSTS(*NOCHG) OUTDTATYP(*ALL) PRTERLOG(*NO) DTLDOC(*NO)
```

The data is stored in the first member in file MYFILE in library MYLIB. The member is given the same name as the document being printed. The document Interchange Document Profile (IDP) status is not checked before printing, nor is the IDP status changed when the printing is complete. The entire document, including IDP information, is printed, but the error log is not. The document is not deleted when printing is complete.

See Appendix C, "Office Services Output File Considerations," for information about the output file layout.

The following command causes document MYDOC in folder MYFLR to be formatted for printing:

```
PRTDOC DOC(MYDOC) FLR(MYFLR) OPTIONS(*PRTFILE)
PRTFILE(MYLIB/MYPRTFIL)
```

Using the above command, the values for the following print options parameters are taken from printer file MYPRTFIL in library MYLIB:

- Number of copies
- Printer
- Output queue
- Form type
- Delay printing (hold)
- Print quality
- Character set and code page
- Type of page printing (duplex)

The following command causes document MYDOC in folder MYFLR to be formatted for printing at printer P3812I:

```
PRTDOC DOC(MYDOC) FLR(MYFLR) DEV(P3812I) OUTQ(MYLIB/MYOUTQ)
PRTQLTY(*LETTER)
```

The output goes to output queue MYOUTQ in library MYLIB, and the printed output is in letter quality.

The following command causes document MYDOC in folder MYFLR to be formatted for printing at the printer stored with the document in its print options:

```
PRTDOC DOC(MYDOC) FLR(MYFLR) JOBQ(*NO)
```


The output also goes to the output queue stored in the print options. The print request is not submitted to the job queue.

The following command causes document MYDOC in folder MYFLR to be formatted for printing at the printer stored with the document in its print options:

```
PRTDOC DOC(MYDOC) FLR(MYFLR) MRGTYPE(*QRY) QRYDFN(MYLIB/MYQRY)
```

The output also goes to the output queue stored in the print options. All Data Field text instructions found during the formatting that have *PRINT as the Merge Source will use query MYQRY in library MYLIB as the merge data source.

Rules

When using the PRTDOC command with the OPTIONS(*PRTFILE) parameter, there are a limited number of the printer file parameters that are actually not used in the printing process. These parameters have corresponding values in the document content itself, and therefore, the document values take precedence. These parameters are listed below:

- CPI
- DRAWER
- FONT
- FORMFEED
- PAGESIZE
- JUSTIFY
- LPI
- OVRFLW
- PAGRTT
- PRTTXT

The following AS/400 document types can be processed using PRTDOC:

- RFTDCA
- FFTDCA
- RFTAS400
- FFTAS400
- RFTS38

If you use OUTPUT(*) to request that the document be printed to the display, the following parameters are ignored:

- PRERRLOG
- COPIES
- JOBQ
- OUTQ
- SPLFILE
- FORMTYPE
- PRTQLTY
- HOLD
- ERRFORM
- COVERPAGE
- DUPLEX
- AUTOBIND

If you use OPTIONS(*OUTFILE) to request that the document be printed to a file, the following parameters are ignored:

- COPIES
- OUTQ
- SPLFILE
- FORMTYPE
- PRTQLTY
- HOLD
- ERRFORM
- COVERPAGE
- DUPLEX
- AUTOBIND

The values specified for the STRPAGE, ENDPAGE, and PAGERANGE parameters are assumed to be the page numbers of the output formatted for printing. If you specify ADJPAGES(*NO) and RENUMBER(*NO), the printed system page numbers closely match your source document. If you specify ADJPAGES(*YES) or ADJPAGES(*NO) RENUMBER(*YES), the page numbers of the source document may not match the page numbers of the output document, especially if the source document contains *includes* or *data* fields. If your source document does not contain *includes* or *data* fields, you can use the F16 key to paginate the source document. This matches the source and output pages in most cases.

If you are printing an FFTAS400 document (and sometimes a FFTDCA document) and you specify a page range that does not exist for the STRPAGE and ENDPAGE or PAGERANGE parameters, an error message is issued stating the page range does not exist and the document is not printed. If you are printing any other type of document, an error message is placed in the error log stating the page range does not exist and the entire document is printed.

The CNLERR parameter of the PRTDOC command allows you to cancel printing if an error is detected during the process of formatting a document. For example, if the document being printed contains an *include* instruction that cannot be processed for some reason (document does not exist, user is not authorized to document) and CNLERR(*YES) is specified, message OFC8E50 is signaled by the PRTDOC command. To determine the cause of the error, the error log for the document must be printed.

Most of the parameters of PRTDOC have an allowed value of *SAME, which uses the value stored in the print options of the document being printed. For example, if you specify ADJPAGES(*SAME), whether or not to adjust page endings is decided based on a value stored in the document print options.

The print options stored with a document can be changed by first editing the document (EDTDOC). When exiting, specify Y for the *Save document* prompt and Y for the *Display print options* prompt. Change any of the values on the Print Options displays, and press the Enter key to end. Any changes made to the print options are stored in the document.

Hints and Techniques

Different types of printers have different printing capabilities. To take advantage of the specific capabilities of a type of printer, the printer data stream created when a document is printed is based on that type of printer. Therefore, the type of the printer chosen on the DEV parameter determines the format of the output in the spooled output file.

Since these printer data streams are type specific, redirection of PRTDOC output from the target printer to another type of printer is not recommended. For example, if DEV(P5219) is specified and P5219 is a model 5219 printer, the output created should not be redirected to printer P5224 where P5224 is a model 5224 printer.

If you anticipate using PRTDOC to change between merge data sources (for example, use the same shell document to merge using several different queries), you should specify *PRINT as the Data Field Source on the Data Field text Instruction. *PRINT means the merge source specified at print time is used.

If the document you are printing contains multiple letters data field text instructions, the source for the document being printed is processed once for each record in your merge data source. It is important to understand this concept as it relates to the Set text instruction. In this case, the Set text instruction can be used to keep a count of the number of records processed (Set RECNUM = RECNUM + 1). It can also be used to determine when the contents of a field change to control the printing of certain information.

For example, if you want to print subtotals each time the value of a field called STATE changes, you can use three Set instructions to accomplish this. The first Set instruction contains the STATE value for the previous record, the second for the current record. The third Set instruction is used to accumulate the subtotal value. The Begin Conditional Text instruction can be used to compare the variables set with the first two Set instructions and cause the subtotals (the third Set variable) to print if the first two values are different.

There are special values allowed on the *Test value* prompt of the Begin Conditional Text (BCT) instruction that cause text to be printed for only the first or the last record in the merge source. These special values are *FIRST and *LAST. For example, you could use the Set instruction to accumulate a total value. You want this total to print when all records have been processed. This can be done by using the special value of *LAST for the test value on the BCT instruction.

There are two methods of placing a PRTDOC request on the job queue:

- Use the SBMJOB command.
- Type JOBQ(*YES) with the PRTDOC command.

Character Set and Code Page Considerations

See Appendix A, "Character Set and Code Page Considerations," for word processing character set and code page considerations.

Print Stop Word List (PRTSWL) Command

The Print Stop Word List (PRTSWL) command prints the words included in an IBM-supplied or user-created stop word list.

Security Considerations

This command is shipped with *USE authority. The object's authority determines if you have authority to print the stop word list.

Some Ways to Use This Command

The following command prints the IBM-supplied stop word list for U.S. English:

```
PRTSWL LANGID(ENU) TYPE(*IBM)
```

Retrieve Stop Word List Source (RTVSWLSRC) Command

The Retrieve Stop Word List Source (RTVSWLSRC) command retrieves the words from an IBM-supplied or user-created stop word list into a source file.

Security Considerations

This command is shipped with *USE authority. The object's authority determines if you have authority to retrieve the stop word list.

Some Ways to Use This Command

The following command retrieves the IBM-supplied stop word list source for U.K. English:

```
RTVSWLSRC LANGID(ENG) SRCFILE(MYLIB/MYFILE) SRCMBR(MYMBR)
```

After you retrieve the stop word list, you can edit the source file to include words you want added or removed from the stop word list. See "Stop Word Lists" on page 7-13 for more information and an example of using the CRTSPADCT command to create a new stop word list.

Start Word Processing (STRWP) Command

The Start Word Processing (STRWP) command displays the Word Processing menu, where you can select from five different functions, or it automatically performs any of those five functions and allows you to bypass the Word Processing menu. For additional detailed information on the STRWP command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

This command is shipped with *USE authority, which means all users can use the command.

Some Ways to Use This Command

Enter the STRWP command from any command line to bypass the OfficeVision/400 menu and go directly to the Word Processing menu.

You can also use the STRWP command to bypass both the OfficeVision/400 menu and the Word Processing menu and go directly to any of the five word processing functions available on the Word Processing menu.

For example, the following command bypasses the OfficeVision/400 menu and the Word Processing menu, and takes you directly to the Work with Documents in Folders display, which is option 1 on the Word Processing menu:

```
STRWP OPTION(1)
```

The following command bypasses the OfficeVision/400 menu and the Word Processing menu, and takes you directly to the Work with Text Profiles display, which is option 5 on the Word Processing menu:

```
STRWP OPTION(5)
```

The STRWP command uses the OPTION parameter. It specifies the option to be selected from the Word Processing menu. Following are the possible values:

***SELECT** The Word Processing menu is displayed so you can select an option from the menu.

- 1 Work with Documents in Folders
- 2 Work with Documents to be Printed
- 3 Work with Folders
- 4 Work with Nontext Document Data
- 5 Work with Text Profiles

Work with Documents (WRKDOC) Command

The Work with Documents (WRKDOC) command takes the user to the Work With Documents in Folders display or the Work with Nontext Document Data display, where several document-related functions can be performed. Using this command allows the user to bypass the OfficeVision/400 menu and the Word Processing menu. Depending on the value specified for the DOC parameter, this command performs the same function as selecting option 1 or 4 from the Word Processing menu.

Security Considerations

This command is shipped with *USE authority, which means all users can use the command.

Some Ways to Use This Command

WRKDOC can be entered from any command line to bypass the OfficeVision/400 menu and the Word Processing menu and go directly to the Work with Documents in Folders display.

This command can be entered from any command line to bypass the OfficeVision/400 menu and the Word Processing menu and go directly to the Work with Nontext Document Data display.

The user can have an option on a user application menu that states Work with documents. Choosing that option calls the WRKDOC command, and takes the user directly to the Work with Documents in Folders display.

The user can have an option on a user application menu that states Work with charts and images. Choosing that option calls the WRKDOC DOC(*NONTXTDTA) command, and takes the user directly to the Work with Nontext Document Data display.

The following command takes the user to the Work with Documents in Folders display, where all the documents in the folder SMITH are shown:

```
WRKDOC DOC(*ALL) FLR(SMITH)
```

The following command takes the user to the Work with Nontext Document Data display, where all the documents in the folder JONES that contain nontext document data (charts, images) are shown:

```
WRKDOC DOC(*NONTXTDTA) FLR(JONES)
```

The following command takes the user to the Work with Documents in Folders display, where all the documents in the folder SMITH are shown. Since the DOC parameter is not specified, it takes *ALL as its default.

```
WRKDOC FLR(SMITH)
```

The following command takes the user to the Work with Documents in Folders display, where all the documents in the folder last used by the user are shown:

```
WRKDOC
```

Rules

If the folder name is not specified when using the WRKDOC command, the folder the user last used is automatically used. Therefore, unless the user works in several different folders, simply typing WRKDOC should show the user the folder to work with.

Hints and Techniques

If currently at the Work with Documents in Folders display, the user can quickly go to the Work with Nontext Document Data display by pressing F9 (Work with ...), then selecting option 4.

If currently at the Work with Nontext Document Data display, the user can quickly go to the Work with Documents in Folders display by pressing F9 (Work with ...), then selecting option 1.

Parameters

The WRKDOC command uses the following parameters:

| | |
|------------|--|
| DOC | Specifies which display to show. Possible values: |
| *ALL | The Work with Documents in Folders display is shown. |
| *NONTXTDTA | The Work with Nontext Document Data display is shown. On this display, you can work with nontext data such as charts or images. |
| FLR | Specifies the name of the folder to be used on the Work with Documents in Folders display or Work with Nontext Document Data display. Possible values: |
| *PRV | The name of the folder from your last session is used. |
| *SELECT | A list of folders is displayed from which you can select a folder. |

Folder name Specify the folder you want to work with on the specified display.

Work with Document Print Queue (WRKDOCPRTQ) Command

The Work with Document Print Queue (WRKDOCPRTQ) command shows the Work with Documents to Be Printed display, from which the user can manage printed output. From this display, the user can change, hold, delete, or release printed output that is waiting to be printed on the output queue and job queue. The user can easily look at subsets of the printed output. For instance, the user can choose to look at output for a specific printer or output queue, or for a specific user. The user can also choose to look at all types of printed output, or only documents printed by OfficeVision/400 word processing.

Security Considerations

This command is shipped with *USE authority, which means all users can use the command.

Some Ways to Use This Command

The user can enter this command on any command line to bypass the OfficeVision/400 menu and the Word Processing menu, and go directly to the Work with Documents to Be Printed display. This command has no parameters.

The user may have an option on a user application menu that states Work with printed output. Choosing that option calls the WRKDOCPRTQ command and takes the user directly to the Work with Documents to Be Printed display.

Rules

The WRKDOCPRTQ command takes you to the Work with Documents to Be Printed display where, among other functions, the user can select option 2 to change a document waiting to be printed. One of the items that can be changed is the printer to which the document is sent for printing. The user must be very careful when changing the printer, since the document prints correctly only if the printer specified on the Change Options display is the same type of printer the document was originally printed on.

The WRKDOCPRTQ command remembers the last subset options specified and shows that subset each time the command is used. For instance, if you divide your list of printed output (using F10) to only show output for printer P3812, then each time you use the WRKDOCPRTQ command you will see that same subset of the printed output.

Hints and Techniques

It is a good idea to divide your list in some way by using F10 (for example, by user, printer, or both). Otherwise, the list of printed output can be extremely large and the system will take longer to show the list each time the WRKDOCPRTQ command is used.

If you are printing a very large document (for example, a large mass mailing), it may take a long time for the document to actually start printing. One reason for this delay is that, by default, nothing prints until the entire document has been converted into printed format. If you are running a mass mailing that includes 1000 one-page letters, all 1000 must be converted before anything appears at the printer. The Change option on the Work with Documents to Be Printed display can be used

for output like this, so printing can begin as soon as the first letter is converted. By changing the Start printing option on the Change Options display to 3 (Immediately), this long delay can be avoided.

Work with Folders (WRKFLR) Command

The Work with Folders (WRKFLR) command shows the Work with Folders display. On this display, the user can select options to create, display folder paths, delete, rename, change the details, work with documents, or change authority to a folder. Using this command allows the user to bypass the OfficeVision/400 menu and the Word Processing menu, and go directly to the Work with Folders display.

Security Considerations

The user must have at least *USE level authority for the folder to work with folders within that folder. This command is shipped with *USE authority, which means all users can use the command.

Some Ways to Use This Command

This command can be entered from any command line to bypass the OfficeVision/400 menu and the Word Processing menu and go directly to the Work with Folders display.

The user can have an option on a user application menu that states Work with folders. Choosing that option calls the WRKFLR command, and takes the user directly to the Work with Folders display.

The following command displays all first-level folders on the Work with Folders display. By not specifying the FLR parameter, the command uses the default, which is *ALL.

```
WRKFLR
```

The following command displays all those folders that reside in the folder DEPTS. These folders are considered second-level folders, since they reside in a first level, or root, folder.

```
WRKFLR FLR(DEPTS)
```

The following command displays all those folders that reside in the second-level folder DEPT800:

```
WRKFLR FLR('DEPTS/DEPT800')
```

Since DEPT800 is a second-level folder, the first-level folder that it resides in (DEPTS) must be included in the folder name. The folders that are displayed by this particular command are considered third-level folders.

Rules

The Work with Folders command always displays those folders that reside in the folder specified for the FLR parameter. If no folder is specified for the FLR parameter, then all first-level folders are displayed.

When displaying first-level files, only those folders that the user has access to are shown on the Work with Folders display. Folders that the user has been excluded from (authority level = *EXCLUDE) are not shown in the list. When displaying second, third, or other level folders, folders from which the user has been excluded

are displayed in the list. However, the user cannot view the folder contents. The folder description displays *NOT AUTHORIZED.

Hints and Techniques

Using the Work with Folder command is the only way the user can specify that a text profile be assigned to a folder. Assigning a text profile to a folder sets a standard document format for each document created in that folder. A text profile can be assigned to a folder from the Work with Folders display when creating a folder (option 1), or when changing the folder details (option 8).

WRKFLR uses the FLR parameter. It specifies the name of the folder used on the Work with Folders display. Possible values are:

| | |
|--------------------|---|
| *ALL | A list that consists of all first-level folders is displayed. |
| Folder name | Only the folders contained in the specified folder are displayed. |

Work with Text Profiles (WRKTXTPRF) Command

The Work with Text Profiles (WRKTXTPRF) command shows the Work with Text Profiles display, where the user can create, revise, copy, delete, or activate text profiles. By using this command the user can bypass the OfficeVision/400 menu and the Word Processing menu, and go directly to the Work with Text Profiles display. For additional detailed information on the WRKTXTPRF command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

This command is shipped with *USE authority, which means all users can use the command.

Some Ways to Use This Command

This command can be entered from any command line to bypass the OfficeVision/400 menu and the Word Processing menu and go directly to the Work with Text Profiles display. This command has no parameters.

The user can have an option on a user application menu that states Work with text profiles. Choosing that option calls the WRKTXTPRF command and takes the user directly to the Work with Text Profiles display.

Rules

The user must have OfficeVision/400 loaded on the system to use this command.

A system profile, called SYSTEM, is provided for OfficeVision/400 word processing users. This text profile is stored in the folder QPRFFLR, and cannot be deleted from the system.

All user text profiles created using the WRKTXTPRF command are stored in the folder QPRFFLR.

Hints and Techniques

Since the system profile (SYSTEM) contains the defaults for all documents created on the system (unless user text profiles are activated), this text profile document should be secured so users cannot make changes to it. For example, set the authority for this document to *PUBLIC *USE, which means that everyone can use it, but only specified users can change it.

Text profiles are nothing more than documents stored in a special folder (QPRFFLR). If you want your text profile to contain some constant text, either in the body of the document or in the header or footer, simply change that profile, specifying QPRFFLR as the folder, and add the text. You can change any text profile document by using the EDTDOC command or by going to the Work with Documents in Folders display.

If you already have a standard shell document set up, and you want to use that document as a text profile, simply copy the document into the folder QPRFFLR. Then activate that text profile document by using option 5 (activate) on the Work with Text Profiles display.

Character Set and Code Page Consideration

When a text profile is created (option 1), the new profile's document format is updated to reflect the character set and code page of the work station device currently being used.

Example Applications

The pages that follow contain examples of using word processing commands to build applications.

Creating Fill Form Document

This example shows one application the user could design using the form field function. The application uses the information from a user database file as the initial values in the FILLFORM command, optionally saves the completed form (shell) document, and uses any changes made in the form document to update the user database file. The objects and document required to run the application are listed below.

| Name | Type | Library | Folder | Description |
|---------|----------|---------|---------|---------------------------|
| ---- | ---- | ----- | ----- | ----- |
| PRSETUP | CL-PGM | FORMAPP | | Set up environment |
| PRSGEN | RPG-PGM | FORMAPP | | Prompt user, call FILLFRM |
| PTNFLE | FILE-PF | FORMAPP | | Patient database file |
| PTNDSPF | FILE-DSP | FORMAPP | | Display file |
| PTNSAVF | FILE-PF | FORMAPP | | FILLFORM output file |
| PTNLF | FILE-LF | FORMAPP | | Logical file to PTNSAVF |
| FILLFRM | CL-PGM | FORMAPP | | Run FILLFORM command |
| PRIFORM | DOCUMENT | | MEDICAL | Shell document |

The following is the code for the PRSETUP program:

```

/*****/
/*
/* CL OBJECT:          PRSETUP          */
/* SOURCE FILE:       QCLSRC            */
/* SOURCE LIB:        FORMAPP           */
/*
/* DESCRIPTION: This program sets up the environment */
/* required to run the application. The library */
/* FORMAPP is added to the library list. The file */
/* PTNSAVF is created by duplicating the system output */
/* file QAOEFDTA. The new file PTNSAVF is then changed*/
/* to enable records to be deleted and reused. This */
/* ensures the file does not get so large that perform-*/
/* ance is degraded. The logical file PTNLF is then */
/* created to access the PTNSAVF file for the changed */
/* information when updating the PTNFLE in the program */
/* PRSGEN. The job attributes are retrieved to be used*/
/* as the key to read the PTNSAVF file. The main */
/* program is then called.
/*
/* *****/

```

Figure 8-2 (Part 1 of 9). Printing All Documents in a Folder

```

|          PGM
|          DCL          VAR(&WRKUSR) TYPE(*CHAR) LEN(10)
|          DCL          VAR(&WRKJOB) TYPE(*CHAR) LEN(10)
|          DCL          VAR(&WRKJB#) TYPE(*CHAR) LEN(6)
|          MONMSG      MSGID(CPF2103 CPF2130 CPF2105)
|          ADDLIBLE    LIB(FORMAPP)
|          DLTF        FILE(FORMAPP/PTNLF)
|          DLTF        FILE(FORMAPP/PTNSAVF)
|          CRTDUPOBJ   OBJ(QAOEFDTA) FROMLIB(QOFC) +
|                    OBJTYPE(*FILE) TOLIB(FORMAPP) NEWOBJ(PTNSAVF)
|          CHGPF      FILE(FORMAPP/PTNSAVF) REUSEDLT(*YES)
|          CRTLF      FILE(FORMAPP/PTNLF) SRCFILE(FORMAPP/QDSSRC) +
|                    SRCMBR(PTNLF)
|          RTVJOBA    JOB(&WRKJOB) USER(&WRKUSR) NBR(&WRKJB#)
|          CALL      PGM(PRSGEN) PARM(&WRKUSR &WRKJOB &WRKJB#)
|          ENDPGM

```

=====

The following is the code for the PRSGEN program:

```

|          H* RPG OBJECT:          PRSGEN
|          H* SOURCE FILE:        QRPGRSRC
|          H* SOURCE LIB:         FORMAPP
|          H*
|          H* DESCRIPTION: Display a screen to prompt for the patient
|          H* number and save indicator. The patient number is used to
|          H* read the database file. The information from the database
|          H* file is passed to the CL program. When control is
|          H* passed back to this program, the information that was
|          H* entered in the form (shell) document that was saved in the
|          H* PTNSAVF database file is used to update the user database file.
|          H* This cycle is repeated until the user presses F3 or F12.
|          H*

```

Figure 8-2 (Part 2 of 9). Printing All Documents in a Folder

```

H*                INDICATOR USAGE
H*                -----
H*    02 - DISPLAY ERROR MESSAGE FOR PATIENT RECORD NOT FOUND
H*    03 - EXIT ONDISPLAY WAS REQUESTED
H*    10 - PTNFLE RECORD NOT FOUND
H*    12 - CANCEL ON DISPLAY WAS REQUESTED
H*    20 - PTNSAVF RECORD NOT FOUND
H*    88 - DISPLAY ERROR MESSAGE FOR UNSUCCESSFUL UPDATE
H*    89 - DISPLAY MESSAGE FOR SUCCESSFUL UPDATE
H*
H*    File name      Description
H*    -----
H*    PTNFLE         User file - Patient database file
H*    PTNDSPF        User display file
H*    PTNLF          Logical file to PTNSAVF-FILLFORM output file
H*
FPTNFLE UF E        K        DISK
FPTNDSPF CF E       K        WORKSTN
FPTNLF  UF E        K        DISK
C*
C* Parameter list received from PRSSETUP
C*
C      *ENTRY      PLIST
C              PARM          WRKUSR 10      user
C              PARM          WRKJOB 10      name of job
C              PARM          WRKJB#  6      job number
C*
C* Build key to be used to read PTNSAVF - FILLFORM output file
C*
C              MOVE'PRSF' WRKDOC 12
C      LFKEY      KLIST
C              KFLD          WRKDOC          document
C              KFLD          WRKUSR
C              KFLD          WRKJOB
C              KFLD          WRKJB#
C              MOVE '0'      ENDPGM          end of pgm
C              Z-ADD99      PRS#    70      prescription
C*
C* The following DO WHILE runs as long as the user does not
C* press F3 or F12. FLD002 is the save indicator that will be
C* passed to the CL program to determine if the completed form
C* document should be saved in the save document specified on the
C* FILLFORM command. The display file appears. The user
C* enters the patient number and saves the choice. If F3 or F12
C* was pressed by the user, the end of program variable is set to 1.
C* The DO WHILE will not run again, the last record indicator
C* is set, and the program ends.
C*

```

Figure 8-2 (Part 3 of 9). Printing All Documents in a Folder

```

C          ENDPGM    DOWEQ'0'
C          MOVE 'N'      FLD002      set variable
C          EXFMTPROMPT  display
C          *IN03    IFEQ '1'      exit
C          *IN12    OREQ '1'      cancel
C          MOVE '1'      ENDPGM 1    set variable
C*
C* The patient database file is read using the user's input.
C* If the patient number is not found, indicator 10 is set on.
C*
C          ELSE
C          FLD001    CHAINPTNFLE      10
C*
C* The patient number is found. The prescription number is
C* incremented. The CL program is called and data passed using
C* the parameter list.
C*
C          *IN10    IFEQ '0'      record found
C          ADD 1      PRS# 70      increment
C          CALL 'FILLFRM'      CL program
C          PARM      FLD002      save?
C          PARM      PRS#      prescription
C          PARM      PTNNBR     patient #
C          PARM      PTNNME     patient name
C          PARM      PTNAD1     address
C          PARM      PTNAD2     address
C          PARM      PTNZIP     zip
C          PARM      PTNDR      doctor
C          PARM      MSGID 7     message id
C*
C* If the data was not written to the PTNSAVF database file,
C* the user database file PTNFLE will not be updated.
C*
C          MSGID    IFEQ 'OFC8DC7'
C          SETON      88      NO UPDATE
C          ELSE
C          SETON      89      UPDATE
C*
C* If the data was written to the PTNSAVF database file, the file
C* is read. If matches are found, the user patient database file
C* is updated. The record in the PTNSAVF file is then deleted
C* and the next record read. The patient database file is
C* updated, the display field set to zero, and the main loop
C* repeated.
C*
C          LFKEY    CHAINPTNLF      20
C          *IN20    DOWEQ'0'
C          FDFFID   IFEQ 'PTNME'
C          MOVELFDDATA PTNNME      patient name
C          END
C          FDFFID   IFEQ 'PTAD1'
C          MOVELFDDATA PTNAD1     address
C          END

```

Figure 8-2 (Part 4 of 9). Printing All Documents in a Folder

```

|           C           FDFPID   IFEQ 'PTAD2'
|           C           MOVELFDDATA   PTNAD2           address
|           C           END
|           C           FDFPID   IFEQ 'ZIP'
|           C           MOVELFDDATA   PTNZIP           zip
|           C           END
|           C           DELETPTNLF
|           C           LFKEY   READEPTNLF           20
|           C           END           IN20 DOWEQ
|           C           UPDATPTNRCD
|           C           END           update good
|           C           Z-ADD0           FLD001
|           C*
|           C* Indicator 10 is on (database record not found), set on the
|           C* indicator 01 to inform the display file to display an error
|           C* message informing the user the patient number was not found.
|           C*
|           C           ELSE
|           C           SETON           02
|           C           END           IN20 IFEQ
|           C           END           IN03&IN12 IFEQ
|           C           END           ENDPGM DOWEQ
|           C           SETON           LR

```

=====

The following is the record format for the physical file PTNFLE:

```

|           A** DDS OBJECT:           PTNFLE
|           A** SOURCE FILE:           QDDSSRC
|           A** SOURCE LIB:           FORMAPP
|           A** DESCRIPTION:           EXAMPLE OF USER DATABASE USED IN PRSGEN
|           A*
|           A           UNIQUE
|           A           R PTNRCD
|           A           PTNNBR           5P 0
|           A           PTNNME           25A
|           A           PTNAD1           25A
|           A           PTNAD2           25A
|           A           PTNZIP           5P 0
|           A           PTNDR           25A
|           A           K PTNNBR

```

=====

Figure 8-2 (Part 5 of 9). Printing All Documents in a Folder

The following is the record format for the display file PTNDSPF:
 Note-When creating the display file (CRTDSPF) use *YES for the
 parameter RSTDSP so the full display will be restored when
 returning from the CL program to the RPG program.

```

A** DDS OBJECT:      PTNDSPF
A** SOURCE FILE:     QDDSSRC
A** SOURCE LIB:      FORMAPP
A** DESCRIPTION:     EXAMPLE OF DISPLAY FILE USED IN PRSGEN
A*
A          R PROMPT                WINDOW(5 20 7 40)
A                                CF03(03) CF12(12)
A                                1 10'PRESCRIPTION GENERATER'
A                                DSPATR(HI)
A                                3 5'PATIENT NUMBER. . . .:'
A          FLD001          5Y 0B 3 28AUTO(RAB) EDTCDE(Z)
A 02                                ERRMSG('PATIENT NOT FOUND' 02)
A 88                                ERRMSG('NO UPDATE TO DB' 88)
A 89                                ERRMSG('UPDATE O.K.' 89)
A                                5 5'SAVE FORM (Y/N) . . .:'
A          FLD002          1A B 5 28VALUES('Y' 'N')

```

=====

The following is the record format for the FILLFORM output file PTNSAVF.
 Notice this is the record format for the system output file. Any file
 created that is to be used as the FILLFORM output file must use this
 format for the IBM supplied FILLFORM programs to function properly.

```

A** DDS OBJECT:      QAOEFDTA
A** SOURCE FILE:     QDDSSRC
A** SOURCE LIB:      QQFC
A** DESCRIPTION:     FILE FOR IBM-SUPPLIED SYSTEM OUTPUT FILE
A*
A          R QOEDATA                TEXT('Form field output data file')
A          FDRCID          2A        TEXT('Record identifier = FD')
A                                COLHDG('RECORD' 'ID')
A          FDSYSN          8A        TEXT('System name')
A                                COLHDG('Sys' 'Name')
A          FDUSER          10A       TEXT('User ID')
A                                COLHDG('User' 'ID')
A          FDJOB#          10A       TEXT('Job Name')
A                                COLHDG('Job' 'Name')
A          FDTMSP          13A       TEXT('Date/Time stamp')
A                                COLHDG('Job' '#')

```

Figure 8-2 (Part 6 of 9). Printing All Documents in a Folder


```

| A          COLHDG('Date/Time stamp')
| A          FDORSN      1A      TEXT('Output Reason')
| A          COLHDG('Output Reason')
| A          FDDOC      12A     TEXT('Shell Document')
| A          COLHDG('Shell Document')
| A          FDFPID      5A     TEXT('Form ID')
| A          COLHDG('Form' 'ID')
| A          FDLGTH     3P 0    TEXT('Data length')
| A          COLHDG('Data length')
| A          FDNUM      1A     TEXT('Numeric only')
| A          COLHDG('Numeric only')
| A          FDDCPS     1P 0    TEXT('Decimal positions')
| A          COLHDG('Dec. positions')
| A          FDNEGV     1A     TEXT('Negative value')
| A          COLHDG('Negative value')
| A          FDDATA     78A    TEXT('Form field data')
| A          COLHDG('Form field data')

```

=====

The following is the record format for the display file PTNLF:

```

| A** DDS OBJECT:      PTNLF
| A** SOURCE FILE:    QDDSSRC
| A** SOURCE LIB:     FORMAPP
| A** DESCRIPTION:    LOGICAL FILE FOR PHYSICAL FILE PTNSAVF
| A*
| A      R QOEDATA          PFILE(PTNSAVF)
| A      FDDOC
| A      FDUSER
| A      FDJOB#
| A      FDJOB#
| A      FDORSN
| A      FDDATA
| A      FDFPID
| A      K FDDOC
| A      K FDUSER
| A      K FDJOB#
| A      K FDJOB#
| A      S FDORSN          COMP(EQ 'E')
| A      S FDORSN          COMP(EQ 'R')

```

=====

Figure 8-2 (Part 7 of 9). Printing All Documents in a Folder

The following is the code for the FILLFRM program:

```

/*****
/*
/* CL OBJECT:          FILLFRM          */
/* SOURCE FILE:       QCLSRC           */
/* SOURCE LIB:        FORMAPP          */
/*
/* DESCRIPTION: This program accepts the parameters */
/* passed by PRSGEN and uses them as initial values */
/* when displaying the document PRSFORM in folder   */
/* MEDICAL. The user may update the information in  */
/* the form fields. The data in all the form fields */
/* defined with IDs is written to the file PTNSAVF. */
/* If the save indicator passed is 'Y', the completed */
/* shell document is saved in the document and folder */
/* specified in the FORMDOC and FORMFLR parameters on */
/* the FILLFORM command.                       */
/*
/*****
PGM          PARM(&SAVE &PRS# &NBR &NME &AD1 &AD2 &ZIP +
              &DR &MSGID)
/*****
/* Variable declares
/*****
DCL          VAR(&SAVE) TYPE(*CHAR) LEN(1)
DCL          VAR(&PRS#) TYPE(*DEC) LEN( 7 0)
DCL          VAR(&NBR) TYPE(*DEC) LEN( 5 0)
DCL          VAR(&NME) TYPE(*CHAR) LEN(25)
DCL          VAR(&AD1) TYPE(*CHAR) LEN(25)
DCL          VAR(&AD2) TYPE(*CHAR) LEN(25)
DCL          VAR(&ZIP) TYPE(*DEC) LEN( 5 0)
DCL          VAR(&DR) TYPE(*CHAR) LEN(25)
DCL          VAR(&MSGID) TYPE(*CHAR) LEN(7)
DCL          VAR(&DOC) TYPE(*CHAR) LEN(8) VALUE('P12345 ')
DCL          VAR(&FLR) TYPE(*CHAR) LEN(7) VALUE('MEDICAL')
DCL          VAR(&PRSA) TYPE(*CHAR) LEN(7)
DCL          VAR(&NBRA) TYPE(*CHAR) LEN(5)
DCL          VAR(&ZIPA) TYPE(*CHAR) LEN(5)
/*****
/* Change variables from numeric to alpha
/*****
CHGVAR      VAR(&PRSA) VALUE(&PRS#)
CHGVAR      VAR(&NBRA) VALUE(&NBR)
CHGVAR      VAR(&ZIPA) VALUE(&ZIP)

```

Figure 8-2 (Part 8 of 9). Printing All Documents in a Folder

```

/*****
/* Logic to check save parameter - */
/* N = do not save complete form */
/*****
IF COND(&SAVE *EQ 'N') THEN(DO)
FILLFORM CPYDOC(PRSFORM) +
          CPYFLR(MEDICAL) +
          DSPSTSLNE(*NO) +
          EXITPNL(*NO) +
          ALWRFSH(*YES) +
          INDATA((PRNBR &PRSA) +
                (PTNBR &NBRA) +
                (PTNME &NME) +
                (PTAD1 &AD1) +
                (PTAD2 &AD2) +
                (ZIP &ZIPA) +
                (DR &DR)) +
          OUTDATA(*ALL) +
          OUTFILE(FORMAPP/PTNSAVF) +
          OUTMBR(*FIRST *ADD) +
          OUTOPT(*YES *NO)
ENDDO
/*****
/* Logic to check save parameter - */
/* Y = save completed form in folder/document */
/* specified. Move the prescription number to */
/* positions 2 - 8 of the document name. */
/*****
IF COND(&SAVE *EQ 'Y') THEN(DO)
CHGVAR VAR(%SST(&DOC 2 7)) VALUE(&PRSA)
FILLFORM CPYDOC(PRSFORM) +
          CPYFLR(MEDICAL) +
          FORMDOC(&DOC) +
          FORMFLR(&FLR) +
          REPLACE(*YES) +
          DSPSTSLNE(*NO) +
          EXITPNL(*NO) +
          ALWRFSH(*YES) +
          INDATA((PRNBR &PRSA) +
                (PTNBR &NBRA) +
                (PTNME &NME) +
                (PTAD1 &AD1) +
                (PTAD2 &AD2) +
                (ZIP &ZIPA) +
                (DR &DR)) +
          OUTDATA(*ALL) +
          OUTFILE(FORMAPP/PTNSAVF) +
          OUTMBR(*FIRST *ADD) +
          OUTOPT(*YES *NO)
ENDDO
RCVMSG MSGTYPE(*INFO) RMV(*NO) MSGID(&MSGID)
MONMSG MSGID(CPF9999) EXEC(CHGVAR VAR(&MSGID) +
                           VALUE('0000000'))
ENDPGM

```

Figure 8-2 (Part 9 of 9). Printing All Documents in a Folder

Printing All Documents in a Folder

The following CL program prints all documents in a given folder to a specific printer:

```
PGM          PARM(&FLR &PRT)
/* THIS PROGRAM WILL PRINT ALL THE DOCUMENTS IN THE */
/* FOLDER SPECIFIED IN THE &FLR TO THE PRINTER      */
/* SPECIFIED IN THE &PRT. BOTH THE FOLDER AND      */
/* THE PRINTER SPECIFIED MUST EXIST.                */
/*                                                    */
/* EXAMPLE CALL OF THIS PROGRAM FROM THE COMMAND    */
/* LINE:                                             */
/*                                                    */
/* CALL PRTFLR PARM(DOCFLR *USRPRF)                 */
/*                                                    */
/* WHERE DOCFLR IS THE NAME OF THE FOLDER CONTAINING */
/* THE DOCUMENTS TO BE PRINTED AND THE PRINTER IS   */
/* THE PRINTER SPECIFIED IN THE USER PROFILE       */
/*                                                    */
DCLF          FILE(QSYS/QADSPDOC) RCDfmt(DOCDTL) /* FILE +
CONTAINING DOCUMENTS TO BE PRINTED */
DCL           VAR(&FLR) TYPE(*CHAR) /* VARIABLE FOR +
THE FOLDER PARAMETER */
DCL           VAR(&PRT) TYPE(*CHAR) /* VARIABLE FOR +
THE PRINTER PARAMETER */
/*                                                    */
/* MONITOR FOR OFC8006 IN CASE THE FOLDER DOES NOT */
/* EXIST                                             */
/*                                                    */
```

Figure 8-3 (Part 1 of 2). Printing All Documents in a Folder

```

MONMSG      MSGID(OFC8006) EXEC(GOTO CMDLBL(END))
/*
/* NOW DISPLAY THE FOLDER TO AN OUTFILE
/*
DSPFLR      FLR(&FLR) TYPE(*DOC) OUTPUT(*OUTFILE) +
            OUTFILE(QTEMP/OUTFILE)
/*
/* MONITOR FOR CPF2105 IN CASE THE OUTFILE DOES NOT
/* EXIST
/*
MONMSG      MSGID(CPF2105) EXEC(GOTO CMDLBL(END))
/*
/* OVERRIDE THE SYSTEM DATABASE FILE FOR THE DSPFLR
/* OUTFILE. CHANGE THE OUTFILE TO THE OUTFILE FROM
/* DSPFLR COMMAND.
/*
OVRDBF      FILE(QADSPDOC) TOFILE(QTEMP/OUTFILE)
/*
/* NOW OPEN THE DSPFLR OUTFILE
/*
OPNDBF      FILE(QSYS/QADSPDOC) OPTION(*INP)
/*
/* READ RECORDS FROM THE OUTFILE UNTIL END OF FILE
/* IS REACHED
/*
READF:      RCVF          RCDfmt(DOCDTL)
/*
/* MONITOR FOR END OF FILE MESSAGE
/*
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(DLTFILE))
/*
/* PRINT THE DOCUMENT NAME FOUND IN THE OUTFILE
/*
PRTDOC      DOC(&DOCNAM) FLR(&FLRNAM) JOBQ(*NO)
            OUTPUT(*PRINT) DEV(&PRT)
/*
/* READ ANOTHER RECORD FROM THE OUTFILE
/*
GOTO        CMDLBL(READF)
/*
/* CLOSE AND DELETE THE OUTFILE
/*
DLTFILE:    CLOF          OPNID(QADSPDOC)
            DLTF          FILE(QTEMP/OUTFILE)
/*
END:        RETURN
            ENDPGM

```

Figure 8-3 (Part 2 of 2). Printing All Documents in a Folder

Spell Checking All Changed Documents in a Folder

The following CL program spell checks all documents in a given folder that have been changed on a given date:

```
PGM          PARM(&FLR &DATE)
/* THIS PROGRAM WILL SPELL CHECK ALL THE DOCUMENTS */
/* IN THE FOLDER SPECIFIED IN THE &FLR THAT HAVE */
/* BEEN CHANGED SINCE THE DATE SPECIFIED ON THE */
/* &DATE. */
/*
/* EXAMPLE CALL OF THIS PROGRAM FROM THE COMMAND */
/* LINE: */
/*
/* CALL CHKFLR PARM(DOCFLR 02/01/89) */
/*
/* WHERE DOCFLR IS THE NAME OF THE FOLDER CONTAINING */
/* THE DOCUMENTS TO BE PRINTED AND THE LAST CHANGED */
/* DATA IS '02/01/89'. */
/*
DCLF          FILE(QSYS/QADSPDOC) RCDfmt(DOCDTL) /* FILE +
              CONTAINING DOCUMENTS TO BE PRINTED */
DCL           VAR(&FLR) TYPE(*CHAR) /* VARIABLE FOR +
              THE FOLDER PARAMETER */
DCL           VAR(&DATE) TYPE(*CHAR) /* VARIABLE FOR +
              THE DATE PARAMETER */
/*
/* MONITOR FOR OFC8006 IN CASE THE FOLDER DOES NOT */
/* EXIST */
/*
MONMSG        MSGID(OFC8006) EXEC(GOTO CMDLBL(END))
/*
/* NOW DISPLAY THE FOLDER TO AN OUTFILE */
/*
DSPFLR        FLR(&FLR) TYPE(*DOC) OUTPUT(*OUTFILE) +
              OUTFILE(QTEMP/OUTFILE)
/*
/* MONITOR FOR CPF2105 IN CASE THE OUTFILE DOES NOT */
/* EXIST */
/*
MONMSG        MSGID(CPF2105) EXEC(GOTO CMDLBL(END))
/*
/* OVERRIDE THE SYSTEM DATABASE FILE FOR THE DSPFLR */
/* OUTFILE. CHANGE THE OUTFILE TO THE OUTFILE FROM */
/* DSPFLR COMMAND. */
/*
OVRDBF        FILE(QADSPDOC) TOFILE(QTEMP/OUTFILE)
```

Figure 8-4 (Part 1 of 2). Spell Checking All Changed Documents in a Folder

```

/* */
/* NOW OPEN THE DSPFLR OUTFILE */
/* */
OPNDBF      FILE(QSYS/QADSPDOC) OPTION(*INP)
/* */
/* READ RECORDS FROM THE OUTFILE UNTIL END OF FILE */
/* IS REACHED */
/* */
READF:      RCVF      RCDfmt(DOCDTL)
/* */
/* MONITOR FOR END OF FILE MESSAGE */
/* */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(DLTFILE))
/* */
/* CHECK THE DOCUMENT NAME FOUND IN THE OUTFILE */
/* IF THE LAST CHANGED DATE IS THE SAME AS THE */
/* DATE PASSED IN ON THE &DATE PARAMETER. */
/* */
IF          +
           COND(&LASTREV = &DATE) THEN(CHKDOC +
           DOC(&DOCNAM) FLR(&FLRNAM) OPTION(*SPELL))
/* */
/* READ ANOTHER RECORD FROM THE OUTFILE */
/* */
GOTO        CMDLBL(READF)
/* */
/* CLOSE AND DELETE THE OUTFILE */
/* */
DLTFILE:    CLOF      OPNID(QADSPDOC)
           DLTF      FILE(QTEMP/OUTFILE)
/* */
END:        RETURN
           ENDPGM

```

Figure 8-4 (Part 2 of 2). Spell Checking All Changed Documents in a Folder

Saving Final Form Documents in a Folder

The following CL program saves a final form copy of all documents changed on a given date into another folder. This can be used as a method of saving documents. If your document-naming convention does not use an extension, then this program deletes the actual printed documents from the output queue. However, the DLTSPFLF command does not work on documents that have a name longer than 10 characters. Therefore, if your document-naming convention does allow an extension, then after this program runs you would need to check the output queue and delete any remaining printed document output.

```

PGM          PARM(&FLR &SAVEFLR &DATE)
/* THIS PROGRAM WILL SAVE ALL THE DOCUMENTS IN THE */
/* FOLDER SPECIFIED IN THE &FLR THAT HAVE BEEN   */
/* CHANGED SINCE THE DATE SPECIFIED ON THE        */
/* &DATE TO THE FOLDER SPECIFIED ON THE &SAVFLR.  */
/*                                                */
/* EXAMPLE CALL OF THIS PROGRAM FROM THE COMMAND  */
/* LINE:                                          */
/*                                                */
/* CALL SAVFLR PARM(DOCFLR SAVEFLR '02/01/89')   */
/*                                                */
/* WHERE DOCFLR IS THE NAME OF THE FOLDER CONTAINING */
/* THE DOCUMENTS TO BE SAVED AND THE LAST CHANGED  */
/* DATE IS '02/01/89' AND THE FOLDER TO SAVE INTO IS */
/* ARCHFLR.                                       */
/*                                                */
DCLF         FILE(QSYS/QADSPDOC) RCDfmt(DOCDTL) /* FILE +
          CONTAINING DOCUMENTS TO BE SAVED      */
DCL          VAR(&FLR) TYPE(*CHAR) /* VARIABLE FOR +
          THE FOLDER PARAMETER                  */
DCL          VAR(&SAVFLR) TYPE(*CHAR) /* VARIABLE FOR +
          THE SAVE FOLDER PARAMETER             */
DCL          VAR(&DATE) TYPE(*CHAR) /* VARIABLE FOR +
          THE DATE PARAMETER                    */
/*                                                */
/* MONITOR FOR OFC8006 IN CASE THE FOLDER DOES NOT */
/* EXIST                                           */
/*                                                */
MONMSG      MSGID(OF8006) EXEC(GOTO CMDLBL(END))
/*                                                */
/* NOW DISPLAY THE FOLDER TO AN OUTFILE           */
/*                                                */
DSPFLR      FLR(&FLR) TYPE(*DOC) OUTPUT(*OUTFILE) +
          OUTFILE(QTEMP/OUTFILE)
/*                                                */
/* MONITOR FOR CPF2105 IN CASE THE OUTFILE DOES NOT */
/* EXIST                                           */
/*                                                */
MONMSG      MSGID(CPF2105) EXEC(GOTO CMDLBL(END))
/*                                                */
/* OVERRIDE THE SYSTEM DATABASE FILE FOR THE DSPFLR */
/* OUTFILE. CHANGE THE OUTFILE TO THE OUTFILE FROM */
/* DSPFLR COMMAND.                                  */
/*                                                */
OVRDBF      FILE(QADSPDOC) TOFILE(QTEMP/OUTFILE)

```

Figure 8-5 (Part 1 of 2). Saving All Documents in a Folder


```

/*                                                                    */
/* NOW OPEN THE DSPFLR OUTFILE                                        */
/*                                                                    */
OPNDBF      FILE(QSYS/QADSPDOC) OPTION(*INP)
/*                                                                    */
/* READ RECORDS FROM THE OUTFILE UNTIL END OF FILE                */
/* IS REACHED                                                       */
/*                                                                    */
READF:      RCVF          RCDfmt(DOCCTL)
/*                                                                    */
/* MONITOR FOR END OF FILE MESSAGE                                */
/*                                                                    */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(DLTFILE))
/*                                                                    */
/* PRINT THE DOCUMENT NAME FOUND IN THE OUTFILE                  */
/* IF THE LAST CHANGED DATE IS THE SAME AS THE                   */
/* DATE PASSED IN ON THE &DATE PARAMETER. HOLD THE               */
/* PRINTED OUTPUT, AND SAVE THE RESOLVED OUTPUT.                 */
/*                                                                    */
IF          +
           COND(&LASTREV = &DATE) THEN(PRTDOC +
           DOC(&DOCNAM) FLR(&FLRNAM) SAVOUTPUT(*YES) +
           SAVDOC(&DOCNAM) SAVFLR(&SAVFLR) +
           HOLD(*YES) JOBQ(*NO))
/*                                                                    */
/*                                                                    */
/* DELETE THE SPOOLED FILE FOR THE DOCUMENT JUST                 */
/* PRINTED                                                         */
/*                                                                    */
DLTSPLF     FILE(&DOCNAM) JOB(*) SPLNBR(*ONLY)
/*                                                                    */
/* READ ANOTHER RECORD FROM THE OUTFILE                          */
/*                                                                    */
GOTO        CMDLBL(READF)
/*                                                                    */
/* CLOSE AND DELETE THE OUTFILE                                  */
/*                                                                    */
DLTFILE:    CLOF          OPNID(QADSPDOC)
            DLTF          FILE(QTEMP/OUTFILE)
/*                                                                    */
END:        RETURN
            ENDPGM

```

Figure 8-5 (Part 2 of 2). Saving All Documents in a Folder

Printing Repetitive Letters

The following is a description of the parts of an application to produce repetitive letters.

A repetitive letter application produces a large number of letters in which the only difference between one letter and the next is information such as name, address, telephone number, and so on. A typical repetitive letter application consists of three parts:

- A source of data, typically as an AS/400 database file or query
- A shell document, typically in an AS/400 folder
- A PRTDOC command to cause the actual letters to print

The source of data can be a database file, a query that selects only certain records from one or more files based on selection values, or another document. One letter is created for each record in the data source.

The shell document contains the text that is the same for each letter. It also contains Data Field text instructions to indicate the text that is different for each letter. To be a repetitive letter application, these Data Field instructions must be identified as multicopy Data Fields when they are created in the shell document. The merge source for these Data Field instructions should be set to *PRINT. This means that the merge source is taken from the parameters specified on the PRTDOC command.

The PRTDOC command contains the name of the shell document to be printed, the folder containing the shell document, the type of the source of data (*QRY, *FILE, or *DOC), and the name of the source of data. It is possible to use a different source of data with the same shell document simply by changing the source of data on the PRTDOC command.

Creating Customized Billings

The following is an example of an application to create customized billings. This customized billing application demonstrates the three parts of a repetitive letters application. For each customer, a chart that shows the purchase dollar amounts for the past 12 months is included.

In order to perform this application, a fill-in document must be used as the merge source. Since this fill-in document must contain a Graphic instruction, a row format is recommended.

An example of the row format and a sample record follows:

```
&FNAME
&LNAME
&MIDINIT
&SALUTE
&ADDR1
&ADDR2
&CITY
&STATE
&ZIP
&AMTDUE
&GRAPH

Thomas
Benda
J
Dr
1100 S. Grandview

Dubuque
IA
50081
103.52
*gr(billhist,G,5,3,,Y,)
```

Figure 8-6. A Sample Record in the Row Format

A shell document is also needed. This shell document is used as input to a MRGDOC command. An example of such a shell document follows:

```
&FNAME &MIDINIT. &LNAME
&ADDR1
&CITY, &STATE &ZIP

Dear &SALUTE. &LNAME

Thank you for your payment of last month's bill. The
XYZ utility company appreciates prompt payment.
Charges for the current month are &AMTDUE. Please
remit by the 10th day of the month. For your
information, we have included a chart of your energy
costs for the past year.

&GRAPH

Sincerely,

John Smith
President, XYZ Utility Company
```

Figure 8-7. Shell Document

The final part of this application is a simple CL program to first run a MRGDOC command against a specific merge source, followed by a PRTDOC command to create the printed output, followed by a DLTDLO command to delete the output from the MRGDOC.

```

PGM          PARM(&DOC &FLR)
/* THIS PROGRAM WILL CREATE THE MONTHLY BILLINGS FOR */
/* THE XYZ UTILITY COMPANY.  THE PARAMETERS PASSED */
/* ARE THE NAME OF THE FILL-IN DOCUMENT AND THE */
/* FOLDER CONTAINING THE DOCUMENT.                */
/*                                                 */
/* EXAMPLE CALL OF THIS PROGRAM FROM THE COMMAND */
/* LINE:                                           */
/*                                                 */
/* CALL MONTHBILL PARM(DATADOC DATAFLR)          */
/*                                                 */
DCL          VAR(&DOC) TYPE(*CHAR) /* VARIABLE FOR +
            THE DOCUMENT PARAMETER                */
DCL          VAR(&FLR) TYPE(*CHAR) /* VARIABLE FOR +
            THE FOLDER PARAMETER                  */
/*                                                 */
/*                                                 */
/* MERGE THE SHELL AGAINST THE INPUT MERGE SOURCE */
/*                                                 */
MRGDOC FROMDOC(SHELLMON) FROMFLR(SHELLFLR) +
        TODOC(TEMP) TOFLR(SHELLFLR) MRGTYPE(*DOC) +
        DTADOC(&DOC) DTAFLR(&FLR) JOBQ(*NO)
/*                                                 */
/* PRINT THE DOCUMENT THAT IS THE OUTPUT FROM THE */
/* MERGE                                           */
/*                                                 */
PRTDOC DOC(TEMP) FLR(SHELLFLR) JOBQ(*NO)
/*                                                 */
/* DELETE THE DOCUMENT THAT IS THE OUTPUT FROM THE */
/* MERGE                                           */
/*                                                 */
DLTDLO DLO(TEMP) FLR(SHELLFLR)
/*                                                 */
/*                                                 */
END:         RETURN
            ENDPGM

```

Figure 8-8. Creating Customized Billings

A Sample Customer Response Application

The following example shows how a simple menu and a few short CL programs can be used together to greatly improve the productivity of a group of word processing users. In this example, the users are responsible for responding to a large number of customer letters that are sent to a large mail-order company. Their job involves responding to both complaint and thank-you letters as quickly as possible with personalized letters. The menu and CL programs allow the users to perform specific customer response tasks by simply selecting an option from the Work with Customer Responses menu, shown in Figure 8-9 on page 8-63. Each option of the menu is linked to a specific CL program. These programs edit, copy, merge, and print specific shell, fill-in, and paragraph documents that reside in a special customer response folder. By combining frequently performed word processing

tasks in a series of CL programs like the ones shown in this example, customer responses can be created and printed much faster and with more ease.

```
*****
*   CR                      Work with Customer Responses                *
*                                                                    *
*   Select one of the following:                                       *
*                                                                    *
*       1. Complaints:   Enter customer information                    *
*       2.                Print response letters                      *
*       3.                Produce customized letter                   *
*       4.                Revise paragraphs                           *
*                                                                    *
*       5. Thank-yous:   Enter customer information                    *
*       6.                Print response letters                      *
*       7.                Produce customized letter                   *
*       8.                Revise paragraphs                           *
*                                                                    *
*                                                                    *
*   Selection or command                                             *
*   ===> _____                                                  *
*                                                                    *
*****
```

Figure 8-9. Sample Customer Response Menu

This menu is created using the Menu function of the Screen Design Aid (SDA) licensed program. In this example, the menu name is CR. It can be called from another application or accessed by typing the GO command on any command line. In an example like this, it is more likely called from the initial menu for the user whose job it is to respond to customer letters.

In the definition of the menu, a call to a different CL program is specified for each of the eight menu options. Selecting option 1 calls the CL program CROPT1. Similar calls are made to CL programs CROPT2 through CROPT8 for items 2 through 8.

Following is a description of the CL programs and the documents they access for each of the options on the CR menu.

Option 1: Enter Customer Information

As each customer complaint letter is received, option 1 is selected to enter the customer information into a fill-in document. This option is selected if the particular customer complaint can be answered with one or more of the predefined response paragraphs. Customer information for all complaint letters received on a given day is entered by selecting this option. At the end of the day, option 2 is selected to merge all the customer information with the shell document and print the response letters. The first time this option is selected on a given day, a shell fill-in document is copied to create a daily use fill-in document. Subsequent selections of option 1 simply cause the daily fill-in document to be changed.

Documents: The following documents help you enter customer information:

CRNAMES: This is the shell fill-in document that includes the variables listed in row format. This document is copied into the daily fill-in document the first time option 1 is selected for the day. This document looks like the following:

```
&name
&lname
&title
&addr
&city
&state
&zip
&response
(rcr) <---- Required carrier return
(pe)  <---- Page end (end of document)
```

Figure 8-10. Customer Response Shell Fill-in Document

For this example, the response variable contains an *include* instruction where the pages (paragraphs) to be included are specified. Generally, only one paragraph (page) is included, but it is possible that a complaint may require more than one standard paragraph to adequately respond to the customer's concerns.

CRCNAME.DAY: This is the name of the daily use fill-in document into which the customer information is entered. It is created at the beginning of the day by copying CRNAMES. Each time option 1 is selected, the document is changed. When this document is changed, the Alt+Down keys are pressed immediately to go to the bottom of the page. Then F5 is pressed, and *npr* is typed to recall the variable form from a notepad called CRFORM. After the variable form is recalled, the customer information is entered in the appropriate places. If a particular variable has no data for a customer, that line is left blank. To specify the paragraphs to be included, the cursor is placed on the instruction character of the *include* instruction and the Enter key is pressed. This shows the Include Instruction display, where the complaint paragraphs document name and folder should already be filled in. The pages to be included for the particular customer are typed in the *Pages to include* prompt and the Enter key is pressed.

This document accumulates all customer information for complaint letters during the day and then is deleted by the CL program called by option 2.

CRFORM: This is the name of the notepad document used to copy in the variable form each time information is entered for a customer complaint letter. If certain variables are usually the same for most customers, those values may already be entered into this notepad document and do not need to be typed in each time. This document looks like the following:

```
(rcr) <--- Where name will be typed
(rcr) <--- Where last name will be typed
Ms.(rcr) <--- Where title will be typed (filled with Ms.)
(rcr) <--- Where address will be typed
(rcr) <--- Where city will be typed
(rcr) <--- Where state will be typed
(rcr) <--- Where zip code will be typed
*inc(rcr) <--- Where paragraphs to include are specified
(rcr) <--- Blank line to separate from next customer
(pe) <--- Page end (end of notepad document)
```

Figure 8-11. Customer Response Notepad Document

Notice that an *include* instruction has already been entered on the eighth line for the response variable data. This instruction should specify the complaint paragraph document and folder. The pages to include change based on the response needed for the customer.

CL Program: The purpose of this short CL (CROPT1) program is to take you directly to the daily fill-in document so the customer information can be entered. It monitors for message 0FC8007, Document not found in folder. It first tries to edit the daily fill-in document, CRCNAME.DAY. If the document exists, fine. If not, message 0FC8007 is issued by the system and a copy of the shell fill-in document is performed prior to editing the daily fill-in document. This CL program looks like the following:

```
PGM
/* THIS PROGRAM WILL EDIT THE DAILY CUSTOMER */
/* RESPONSE FILL-IN DOCUMENT FOR CUSTOMER */
/* COMPLAINTS. IF THE DAILY FILL-IN DOCUMENT */
/* (CRCNAME.DAY) DOES NOT EXIST, THEN IT IS */
/* CREATED BY COPYING THE BASE FILL-IN DOCUMENT */
/* (CRNAMES). */
/* */
MONMSG MSGID(0FC8007) EXEC(GOTO CMDLBL(CPY))
/* */
/* MONITOR FOR 0FC8007 IN CASE THE DAILY FILL-IN */
/* DOCUMENT FOR CUSTOMER COMPLAINTS DOES NOT EXIST */
/* */
```

Figure 8-12 (Part 1 of 2). Program to Edit Daily Fill-in Document

```

/* */
/* ATTEMPT TO EDIT THE DAILY FILL-IN DOCUMENT */
/* FOR CUSTOMER COMPLAINTS ... */
/* */
EDTDOC      DOC(CRCNAME.DAY) FLR(CRLIB)
/* */
GOTO        CMDLBL(END)
/* */
/* COPY THE SHELL FILL-IN DOCUMENT INTO THE */
/* DAILY CUSTOMER COMPLAINT FILL-IN DOCUMENT */
/* */
CPY: CPYDOC      FROMDOC(CRNAME) FROMFLR(CRLIB) +
      TODOC(CRCNAME.DAY) TOFLR(CRLIB)
/* */
/* NOW EDIT THE DAILY FILL-IN DOCUMENT */
/* FOR CUSTOMER COMPLAINTS ... */
/* */
EDTDOC      DOC(CRCNAME.DAY) FLR(CRLIB)
/* */
END:        RETURN

```

Figure 8-12 (Part 2 of 2). Program to Edit Daily Fill-in Document

Option 2: Print Response Letters

This option is selected at the end of the day to print the response letters to the customers who were added to the daily fill-in document. Selecting this option causes one response letter to be printed for each set of customer information entered using option 1. After printing the response letters, the daily fill-in document is deleted.

CRCSHL is the shell document used to produce the response letters for customer complaints. It includes data field variables for each of the variables defined in the fill-in document shell. The data source for each of these data field instructions should be set to *PRINT, which means the source of the data is supplied at print time. It also includes a standard closing paragraph. The selected response paragraph(s) are included prior to this closing paragraph. This document looks like the following:

```
*date

*&name
*&addr
*&city, *&state *&zip

Dear *&title *&lname:

*&responseThank you very much for taking the time to write
to us describing your problem. We at XYZ Company believe
your satisfaction with our services is of the utmost
importance. We hope the actions described above will make
up for any inconvenience you may have experienced, and we
hope you will continue to choose XYZ in the future for your
mail order needs.

Sincerely yours,

Mary Johnson
Customer Relations
```

Figure 8-13. Shell Document

The purpose of this CL program (CROPT2) is to print a response letter to each customer whose information was entered into the daily fill-in document using option 1. Selecting option 2 is done at the end of the day. This program first uses the MRGDOC command to include the paragraph(s) specified in the *include* instruction. The program then uses the PRTDOC command to print the response letters, and then deletes the daily fill-in document using the DLTDL0 command. If copies of the letters are to be kept, provisions can be made to this CL program. The CL program looks like the following:

```

PGM
/* THIS PROGRAM WILL PRINT THE RESPONSE LETTERS      */
/* TO THOSE CUSTOMERS IN THE COMPLAINT FILL-IN        */
/* DOCUMENT. THIS IS INTENDED TO BE AN "END OF       */
/* THE DAY" PROCESS. AFTER PRINTING THE RESPONSE     */
/* LETTERS, THE DAILY FILL-IN DOCUMENT FOR CUSTOMER */
/* COMPLAINTS (CRCNAME.DAY) WILL BE DELETED.        */
/*                                                    */
/*                                                    */
/* MONITOR FOR OFC8007 IN CASE THE DAILY FILL-IN     */
/* DOCUMENT FOR CUSTOMER COMPLAINTS DOES NOT EXIST  */
/*                                                    */
MONMSG      MSGID(OFC8007) EXEC(GOTO CMDLBL(END))
/*                                                    */
/* USE THE MRGDOC COMMAND AGAINST THE SHELL          */
/* DOCUMENT TO INCLUDE THE PARAGRAPHS FROM THE      */
/* PARAGRAPH DOCUMENT INTO THE SHELL DOCUMENT.     */
/*                                                    */
MRGDOC      FROMDOC(CRCSHEL) FROMFLR(CRFLR) +
            TODOC(CRCTEMP) TOFLR(CRFLR) +
            REPLACE(*YES) JOBQ(*NO) MRGTYPE(*DOC) +
            DTADOC(CRCNAME.DAY) DTAFLR(CRFLR)
/*                                                    */
/* THEN THE CUSTOMER COMPLAINT SHELL IS PRINTED     */
/*                                                    */
PRTDOC      DOC(CRCTEMP) FLR(CRFLR) MRGTYPE(*DOC) +
            DTADOC(CRCNAME.DAY) DTAFLR(CRFLR)
/*                                                    */
/* THEN THE DAILY FILL-IN DOCUMENT IS DELETED       */
/*                                                    */
DLTDLO      DLO(CRCNAME.DAY) FLR(CRFLR)
/*                                                    */
/*                                                    */
END:        RETURN
            ENDPGM

```

Figure 8-14. Program to Print Customer Response

Option 3: Produce Customized Letter

This option is selected when the inclusion of a standard paragraph or two into the shell document is not an adequate response to a particular customer complaint. In such cases, additions or deletions need to be made after the standard paragraph(s) are included into the shell. Selecting this item allows the user to change the document that results from merging the data from the fill-in document and the specified paragraphs with the shell document. That document, with its changes, is then printed.

Documents: The following documents help you produce a customized letter:

- CRCNAME.CUS:** This is a fill-in document where the customer information for this customized letter is entered. It is created by copying the shell fill-in document CRNAMES.
- CRCCUS1:** This document is the result of the first MRGDOC command. It contains the constant text from the shell document and the data from the fill-in document (CRCNAME.CUS).
- CRCCUST:** This document is the result of the second MRGDOC command. It includes the specified paragraph(s) with the text and data from document CRCCUS1. This is the document that appears at the edit display, so changes can be made. When the customized changes are finished, this document is printed.

CL Program: The purpose of the following CL program (CROPT3) is to merge the customer information, the specified paragraphs, and the shell document to create a new document. This new document, CRCCUST, is then changed using the EDTDOC command so the changes can be made for the customer's specific situation. This CL program looks like the following:

```
PGM
/* THIS PROGRAM WILL PERFORM SEVERAL COMMANDS          */
/* TO ALLOW THE USER TO CUSTOMIZE A RESPONSE           */
/* LETTER FOR A CUSTOMER COMPLAINT LETTER.             */
/* AFTER MERGING THE APPROPRIATE PARAGRAPH(S) WITH    */
/* THE SHELL, THE USER WILL THEN BE ABLE TO EDIT     */
/* THE MERGED RESULT TO MAKE FINAL CUSTOMIZED        */
/* CHANGES.                                           */
/*                                                     */
/* MONITOR FOR OFC8007 IN CASE ANY OF THE DOCUMENTS  */
/* USED IN THIS PROGRAM DO NOT EXIST.                */
/*                                                     */
MONMSG      MSGID(OFC8007) EXEC(GOTO CMDLBL(END))
/*
/*
/* COPY THE SHELL FILL-IN DOCUMENT INTO THE          */
/* CUSTOMIZED COMPLAINT FILL-IN DOCUMENT             */
/*
CPYDOC      FROMDOC(CRNAMES) FROMFLR(CRFLR) +
            TODOC(CRCNAME.CUS) TOFLR(CRFLR) +
            REPLACE(*YES)
/*
/*
```

Figure 8-15 (Part 1 of 2). Program to Produce Customized Response

```

/* NOW EDIT THE CUSTOMIZED FILL-IN DOCUMENT          */
/* FOR CUSTOMER COMPLAINTS ...                        */
/*                                                    */
EDTDOC      DOC(CRCNAME.CUS) FLR(CRFLR)
/*                                                    */
/* NOW MERGE THE FILL-IN DOCUMENT WITH THE SHELL     */
/* DOCUMENT TO CREATE A NEW DOCUMENT THAT CONTAINS  */
/* THE CUSTOMER DATA FROM THE FILL-IN DOCUMENT.    */
/*                                                    */
MRGDOC      FROMDOC(CRCSHEL) FROMFLR(CRFLR) +
            TODOC(CRCCUS1) TOFLR(CRFLR) +
            REPLACE(*YES) JOBQ(*NO) MRGTYPE(*DOC) +
            DTADOC(CRCNAME.CUS) DTAFLR(CRFLR)
/*                                                    */
/* NOW MERGE THE NEWLY CREATED DOCUMENT (CRCCUS1)   */
/* SO THE SELECTED PARAGRAPHS ARE INCLUDED INTO A   */
/* NEW DOCUMENT THAT CAN BE EDITED.                */
/*                                                    */
MRGDOC      FROMDOC(CRCCUS1) FROMFLR(CRFLR) +
            TODOC(CRCCUST) TOFLR(CRFLR) +
            REPLACE(*YES) JOBQ(*NO)
/*                                                    */
/* NOW EDIT THE NEWLY CREATED RESPONSE LETTER SO    */
/* IT CAN BE CUSTOMIZED ...                          */
/*                                                    */
EDTDOC      DOC(CRCCUST) FLR(CRFLR)
/*                                                    */
/* FINALLY, PRINT THE RESULTING CUSTOMIZED LETTER   */
/*                                                    */
PRTDOC      DOC(CRCCUST) FLR(CRFLR)
/*                                                    */
/*                                                    */
END:        RETURN
            ENDPGM

```

Figure 8-15 (Part 2 of 2). Program to Produce Customized Response

Option 4: Revise Paragraphs

This option is selected whenever changes or additions need to be made to the paragraphs that are used to respond to customer complaints.

Documents: CRCPARA is the document that contains the standard response paragraphs for customer complaint letters. Each page of this document contains a separate paragraph that deals with a specific complaint. Each page of this document looks like the following:

```

Xxxxx xxxxx xxxxx xxxxx xxxxx xxxxx xxxxx xxxxx xxx x
xxxx xxxxx xxxxxxxxxxx xxxxxx xxxxx xxxxx xxxxx xxx xx x xxx
xx xxxxx xxxxxx xxxxxx xxxxx xxxxx xxxxxxxxxxx xxx x xxxxx x
xxxx xxxxx xxxxxxxxxxx xxxxxx xxxxx xxxxx xxxxx xxx xx x xxx
xx xxxxx xxxxxx xxxxxx xxxxx xxxxx xxxxxxxxxxx xxx x xxxxx x
xxxx xxxx xxxxxx.(rcr)
(rcr)
(pe)

```

Figure 8-16. Standard Response Document

One blank line must follow the last line of the paragraph. This blank line contains a single required carriage return (rcr). The end of the page (pe) immediately follows the blank line. Each page of the paragraphs document must follow this format.

CL Program: The purpose of the following CL program (CROPT4) is to allow the user to easily edit the document that contains the standard paragraphs used in responding to customer complaint letters. This option is selected whenever additional paragraphs need to be added, or changes need to be made to existing paragraph(s). This CL program looks like the following:

```

PGM
/* THIS PROGRAM WILL EDIT THE DOCUMENT THAT          */
/* CONTAINS THE RESPONSE PARAGRAPHS FOR THE          */
/* COMPLAINT LETTERS THAT ARE RECEIVED              */
/* (CRCPARA).                                         */
/*                                                    */
/* MONITOR FOR OFC8007 IN CASE THE PARAGRAPHS        */
/* DOCUMENT DOES NOT EXIST..                         */
/*                                                    */
MONMSG      MSGID(OF8007) EXEC(GOTO CMDLBL(END))
/*                                                    */
/* ATTEMPT TO EDIT THE PARAGRAPHS DOCUMENT FOR      */
/* COMPLAINT LETTERS ...                             */
/*                                                    */
EDTDOC      DOC(CRCPARA) FLR(CRFLR)
/*                                                    */
/*                                                    */
RETURN
ENDPGM

```

Figure 8-17. Program to Edit the Response

Option 5 through Option 8: Thank You Letters

The CL programs for options 5 to 8 of this application are basically identical to options 1 to 4, except different shell and paragraph documents are accessed. A different shell document is used because the closing paragraph for thank-you letters is different from that of the complaint shell. Also, different response paragraphs are necessary for thank-you letters. A different notepad document is used when typing the customer information into the fill-in documents, since the *include* instruction needs to refer to the thank-you paragraphs document instead of the complaint paragraphs document.

Chapter 9. Data Backup, Recovery, and Storage Management

This chapter contains information to help you save, restore, reorganize, and reclaim individual objects created by the office services described in previous chapters. For information about developing a backup and recovery strategy that includes office data, see the *Basic Backup and Recovery Guide*.

This chapter also discusses how to manage storage used by office data and how you can free storage used by documents by saving them with the storage free option. Also discussed is how to manage journal receivers used by office services.

Figure 9-1 illustrates the two system libraries that contain office services data.

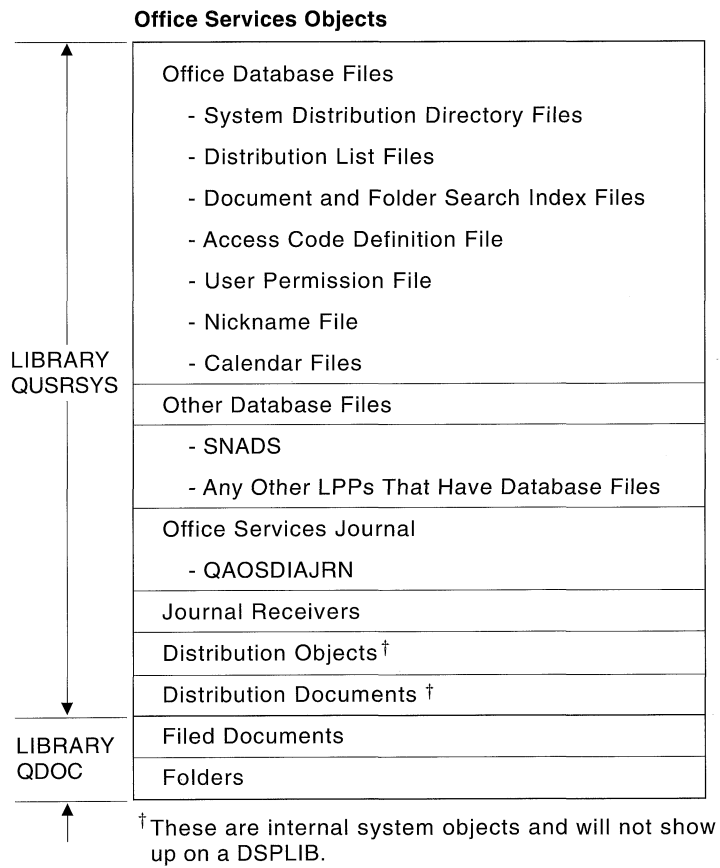


Figure 9-1. Office Services Objects

Figure 9-2 illustrates the structure of mail, changes from the previous versions, and the present mail save and restore support.

Figure 9-2. Saving Mail

| Objects That Make Up Mail | Version 1 and Version 2 Release 1 | | Version 2 Release 2 | |
|--|-----------------------------------|-----------|---------------------|-------------------|
| | Save | Restore | Save | Restore |
| Distribution Objects | SAVSYS or SAVSECDTA | RSTUSRPRF | SAVDLO | RSTDLO (see note) |
| Distribution Documents and Filed Documents | SAVDLO | RSTDLO | SAVDLO | RSTDLO |

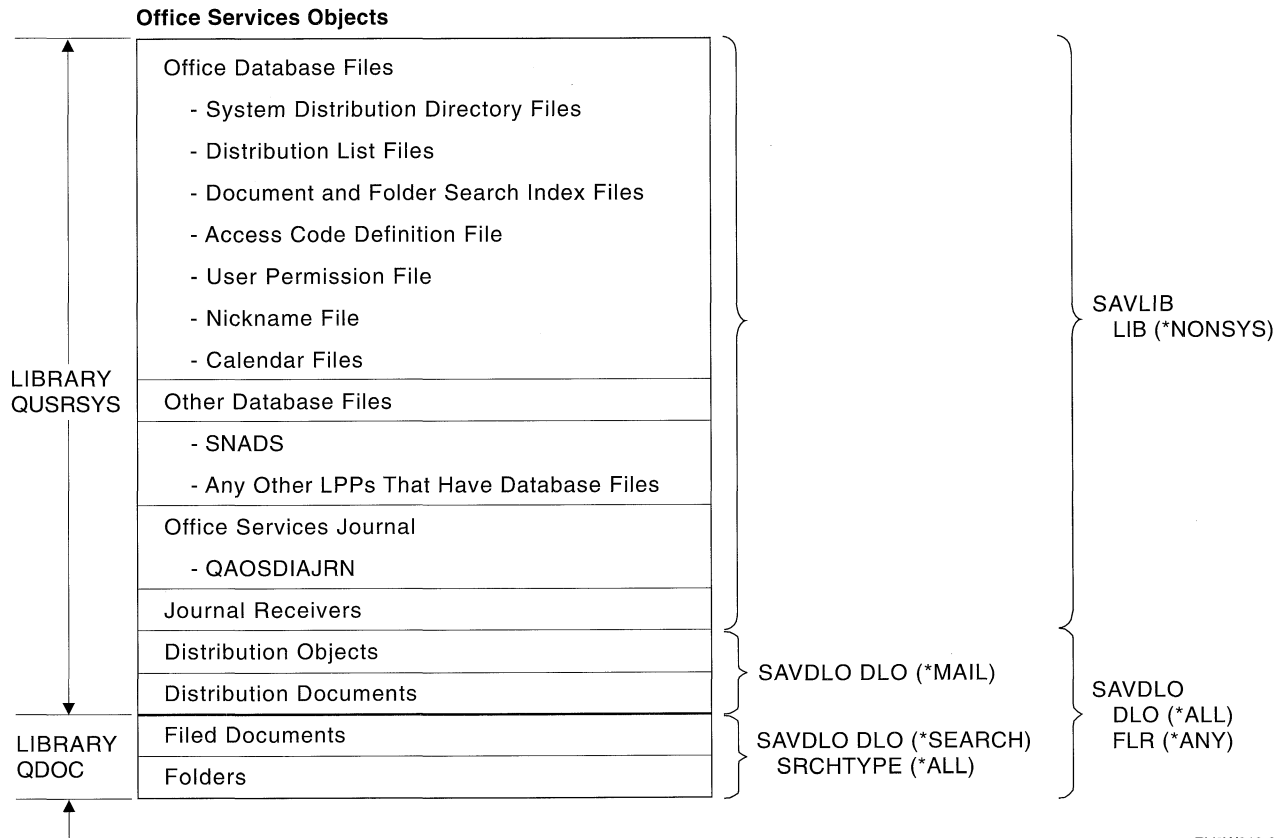
Note: Use the RSTUSRPRF command if the distribution objects were saved before Version 2 Release 2.

Figure 9-3 illustrates the changes from previous versions and the present calendar save and restore support.

Figure 9-3. Calendar Save and Restore Support

| Valid Calendar Save and Restore Support | Version 2 Release 2 Interactive Restore | Version 1 Release 3 Interactive Restore | Version 2 Release 1 Interactive Restore or RSTCAL Command |
|--|---|---|---|
| Version 1 Release 2 Interactive Save | single calendar | single calendar | single calendar |
| Version 1 Release 3 Interactive Save | | single calendar | single calendar |
| Version 2 Release 1 Interactive Save or SAVCAL command | | single calendar | one or many calendars |
| SAVLIB LIB(*NONSYS) | | | one or many calendars |

Figure 9-4 illustrates the overall relationship between system libraries and the save and restore commands.



RV2W210-2

Figure 9-4. Office Services Objects and the Save and Restore Commands

Individual Object Restore

The following describes how to restore individual objects, including individual documents and folders, distribution objects, and database files.

Restore of Individual Documents and Folders

Individual filed documents and folders can be restored from any tape (except a SAVSTG tape), diskette, or save file to which they were saved.

If a filed document is restored individually and it is referred to by a distribution object, the reference to the distribution object is restored as long as the required distribution object still exists on the system. If the required distribution object does not exist on the system, then the filed document is still restored but you will receive an informational message stating that the document was restored but is no longer a part of mail.

Documents and folders can be restored by using a list. Restore a list of documents by entering RSTDLO DLO(A B C D E) FLR(X), where A, B, C, D, and E are documents saved from folder X. If you save a folder you can restore individual documents from it. All the documents in the list that you restore with one RSTDLO command must exist in one diskette file, tape file, or one save file.

Folders can also be restored by using a list. Restore a list of folders by entering RSTDLO DLO(*ALL) SAVFLR(F G H I J) where F, G, H, I and J are folders that must exist in the same tape or diskette file or in the same save file.

You may restore up to 300 documents and folders by entering a list with one RSTDLO command.

Distribution objects and documents cannot be restored individually. The only way you can restore saved distribution objects and documents is by entering RSTDLO DLO(*MAIL) or RSTDLO DLO(*ALL) SAVFLR(*ANY). Using this command to restore distribution objects and documents is intended for disaster recovery purposes only and not as a means of restoring individual mail logs.

A more detailed description of the SAVDLO and RSTDLO commands can be found in the *CL Reference* manual and the *Basic Backup and Recovery Guide*.

Restore of Database Files

You should never restore a previous version physical file unless it has become damaged or destroyed on your system. See "Damaged Database Files" on page 9-13.

Restore of Directory

The directory can only be restored by either the RSTSTG or RSTLIB SAVLIB (QUSRSYS) on the system on which it was saved. Directories cannot be copied to another system because they are system-specific. Use directory shadowing to get directory data on other systems in your network.

Restore of Calendars

Individual calendars can be restored from the complete saves of office data. The only save and restore functions that operate on individual calendars are the Save Calendar and Restore Calendar functions.

You can use these commands or functions for general backup and recovery of calendars, but they are not intended for this purpose. You can also use these commands to move users from one system to another or to restore calendars from a general system backup.

Note: Saving all calendars using the SAVCAL command does not save all calendar files. SAVCAL saves enough data so the calendars can be restored. RSTCAL merges saved items onto the calendars specified.

See the *CL Reference* manual for additional information on these commands.

Reclaim Document Library Object (RCLDLO) Command

The Reclaim Document Library Object (RCLDLO) command is used to reclaim a folder, a document, or internal document library system objects, or synchronize the search index database with the document library.

Security Considerations

You do not need any authority to a document or folder to reclaim it. You do not need to be enrolled in the system distribution directory to use this command. Exclusive use of the object being reclaimed is required.

To reclaim internal system objects or to synchronize the search index files with the document library, you must have *ALLOBJ or *SECADM authority. These functions can only be done when no other jobs are using folders or documents.

Some Ways to Use This Command

The RCLDLO command has three main purposes:

- Rebuild the data structures contained within a document or folder. This can be performed on an individual object basis or the DLO(*FLR) parameter can be used to reclaim a folder and all folders and documents with the folder.
- Rebuild internal document library system objects. This function is started using the DLO(*INT) parameter.

Note: The internal document library system objects are used to manage the documents and folders on the AS/400 system. RCLDLO DLO(*INT) is necessary only if the internal objects become damaged. If the internal objects are damaged, attempts to access documents and folders result in the message CPF8A46 (Internal system objects are damaged), possibly followed by the message CPF9032 (Document interchange session not started).

- Rebuild internal document library system objects and all folders and documents on the system, including synchronizing their document details. This function is started using the DLO(*ALL) parameter.

The *ALL parameter synchronizes all document library objects and text search. This can take a significant amount of system time. If you can issue the RCLDLO command at your discretion, you may want to avoid the operation until the required time can be scheduled.

The following items indicate what happens to folders and documents that are synchronized with the search index database files or have lost folder path information:

- A complete log of the actions taken by the RCLDLO DLO(*ALL) command is sent to the job log, and a list of status messages is sent to the QHST log. Additional information about the objects shown in the log can be displayed by the Display Object Description (DSPOBJD) command.
- Objects that have lost folder path information are associated with a system reclaim folder. An informational message is sent to the QSYSOPR message queue for each object reclaimed in this manner. System reclaim folders, recognized by the name QRCLnnnn.DOC or QRCLnnnn.FLR, should be deleted from the system after you move the objects contained within them to an appropriate user folder. You may also need to use the

Edit Document Library Object Authority (EDTDLOAUT) command to grant specific authority to the reclaimed objects.

The RCLDLO DLO(*ALL) command and the RCLDLO DLO(*INT) command first perform processing that requires no other DLO activity on the system. When this is finished, a CPC8A21 message is sent to the QSYSOPR message queue. After this message is sent, other jobs are allowed to do DLO activity. The RCLDLO command then continues processing documents and folders that need to be reclaimed.

Examples

This command reclaims folder FLR1:

```
RCLDLO DLO(FLR1)
```

This command reclaims folder or document A in folder FLR2:

```
RCLDLO DLO(A) FLR(FLR2)
```

This command reclaims all the folders and documents in folder FLR3:

```
RCLDLO DLO(*FLR) FLR(FLR3)
```

This command reclaims the internal document specified by the system object name AMBT133080:

```
RCLDLO DLO(*SYSOBJNAM) SYSOBJNAM(AMBT133080) SYSOBJATR(*INTDOC)
```

This command reclaims the distribution document specified by the system object name AMBT133082:

```
RCLDLO DLO(*SYSOBJNAM) SYSOBJNAM(AMBT133082) SYSOBJATR(*DST)
```

This command reclaims internal document library system objects:

```
RCLDLO DLO(*INT)
```

This command reclaims internal document library system objects and all documents and folders, and synchronizes the corresponding document search index files:

```
RCLDLO DLO(*ALL)
```

Reorganize Document Library Object (RGZDLO) Command

The Reorganize Document Library Object (RGZDLO) command allows you to reorganize document library object contents, removing unused storage.

Security Considerations

You must have at least *CHANGE authority to the document library object.

You must have *ALLOBJ or *SECADM special authority to specify DLO(*MAIL) or to specify DLO(*ALL) and FLR(*ANY).

Some Ways to Use this Command

Reorganize all document library objects including unfiled mail documents:

```
RGZDLO DLO(*ALL) FLR(*ANY) MAIL(*YES)
```

Reorganize all filed documents that have not been used in seven days and are not contained in any folder:

```
RGZDLO DLO(*ALL) FLR(*NONE) DAYS(7)
```

Reorganize all document library objects contained in a folder named FOLDER1:

```
RGZDLO DLO(*ALL) FLR(FOLDER1)
```

Reorganize all unfiled mail documents:

```
RGZDLO DLO(*ALL) FLR(*ANY) MAIL(*ONLY)
```

Reorganize a document named DOC1 contained in a folder named FOLDER1:

```
RGZDLO DLO(DOC1) FLR(FOLDER1)
```

Rules

Exclusive access is required to reorganize a document or folder.

To reorganize distribution documents, office activity must be stopped.

Hints and Techniques

The RGZDLO command may run for a long time. The duration depends on several factors, among them are:

- The number of document library objects requested for reorganization.
- The values specified on the command.
- The amount and type of activity on the document library objects since last reorganized.
- The other activity on the system while the command is running.

If you want to reorganize all document library objects you may want to run this command in a batch job at a time when other system activity is low.

When a document library object is known to be already reorganized it will not be reorganized again.

The RGZDLO command produces messages that report how much storage was recovered by running the command. You may use this information to determine how often to run this command.

Office Data Storage and Journal Management

There are three areas that need to be managed to control the amount of storage used by office services data. The following objectives should apply:

- Reduce the amount of storage occupied by nontextual document data
- Reduce the amount of storage associated with database files and journaling
- Reduce the amount of storage associated with distribution objects

Limiting the total amount of storage allocated to an individual user can be done using the MAXSTG parameter on the CRTUSRPRF and CHGUSRPRF commands. However, this only affects the creation of objects owned by a specific user profile. In the case of office database files, journals, and distribution objects, the storage is not charged to individual office services users.

Managing Storage Used by Documents and Folders

It is to your advantage to not allow your system to become overcrowded with too many documents, and to not allow your folders to become too large. Many office services functions perform better when your system is not overloaded with documents. The number of documents you keep active on your system should depend on the size of your system, the functions you use, the number of users, and level of performance acceptable to you.

Develop a strategy for saving to offline those storage documents and folders that are not often needed online. One method is to assign an expiration date to your document, such as the end of a month or the end of a quarter. Then you can save all the documents that are past their expiration date and delete them. You can delete them using the DLTDLO (Delete Document Library Object) command, or the STG(*DELETE) keyword on the SAVDLO command. Or, you can use the STG(*FREE) keyword on the SAVDLO command to delete the document text but keep the document details on the system so that the document name still appears in search results. Using STG(*FREE) deletes the document text, which provides more usable space on your system for other objects.

If you do migrations of many documents and folders from a System/36 or System/38 to an AS/400 system, you should check the list of documents and folders on your AS/400 system after the migration for any documents or folders that you do not plan to use often. Save these unused documents and folders to offline storage and delete them from the system.

In the normal use of document library objects, these objects may accumulate storage that is no longer used. Run the RGZDLO command described on “Reorganize Document Library Object (RGZDLO) Command” on page 9-6 to remove this unused storage.

Managing Storage Used by Database Objects and Journals

This section discusses techniques to help you manage the storage used by database objects, journals and distribution objects.

Compressing Database Files

There are a number of database files in the QUSRSYS library that contain office services data. These database files are subject to changes, additions, and deletions. The document and folder search index files can reuse space used by deleted records and do not normally need reorganizing. However, you might want to reorganize them if a large number of documents are deleted and a large number of documents is not expected to be added. Use the Reorganize Physical File Member (RGZPFM) command to reorganize the following files. For more information on the RGZPFM command, see the *CL Reference* manual.

Figure 9-5. Files Associated with the Document and Folder Search Index

| | | |
|----------|----------|----------|
| QAOSSS10 | QAOSSS13 | QAOSSS17 |
| QAOSSS11 | QAOSSS14 | QAOSSS18 |
| QAOSSS12 | QAOSSS15 | |

The directory files also use the reuse deleted records option. You should periodically reorganize the following physical files using the RGZPFM command to reorder the records for performance.

Figure 9-6. Files Associated with Directories, Distribution Lists, and Nicknames

| | | |
|----------|----------|----------|
| QAOKP01A | QAOKP06A | QAOKPLCA |
| QAOKP02A | QAOKP08A | QAOKPLGA |
| QAOKP03A | QAOKP09A | QAOKPSPA |
| QAOKP04A | QAOKPX4A | |
| QAOKP05A | QAOKPCLA | |

The calendar files can accumulate deleted records over time. You should periodically reorganize the following physical files using the RGZPFM command. This reclaims storage used by deleted records.

Figure 9-7. Files Associated with Calendars

| | | |
|---------|---------|---------|
| QAOCAU5 | QAOCIT5 | QAOCMA5 |
| QAOCIN5 | QAOCGM5 | QAOCXT5 |
| QAOCCL5 | QAOCGR5 | |

You can determine the number of deleted records in a physical file by using the Display File Description (DSPFD) command.

Note: Use the RGZPFM parameter on the RMVICALITM command to reorganize the calendar files. This allows you to remove old items and reorganize the files in one step. The following example does not remove any items, but reorganizes the files:

```
RMVICALITM CAL(*ALL *ALL) ITMAGE(99999) RGZPFM(*YES)
```

Use of Commitment Control and Journal Management

Commitment control and journaling are used by directory services and document library services to ensure that changes made to related objects and database files are done automatically. Should an activity resulting in changes to one or more database files and related objects be canceled prior to completing the entire activity, commitment control allows the rollback of any partial changes to the database. Directory services and document library services journal all changes made to directory and document and folder search index database files. The journaling of these files allows for individual recovery of a directory or library services database file in the event of damage or deletion. Commitment control and journaling are used to maintain the integrity and the recoverability of office services database files.

The journal used by directory services and document library services is QAOSDIAJRN and is kept in QUSRSYS. The name of this journal cannot be changed and the journal cannot be removed from the system if office activity is to continue. Office activity includes the use of such products as OfficeVision/400, any remote DIA applications, or PC Support/400. The AS/400 system ships QAOSDIAC01 in library QUSRSYS as the initial journal receiver. All files journaled by QAOSDIAJRN are also in QUSRSYS. The following is a list of those files:

Figure 9-8. Files Journaled to QAOSDIAJRN

| | | |
|----------|----------|---------|
| QAOKP01A | QAOKPX4A | QAOSY11 |
| QAOKP02A | QAOKPCLA | QAOSY12 |
| QAOKP03A | QAOKPLCA | QAOSY13 |
| QAOKP04A | QAOKPLGA | QAOSY14 |
| QAOKP05A | QAOKPSPA | QAOSY15 |
| QAOKP06A | QAOSAH05 | QAOSY17 |
| QAOKP08A | QAOSAH07 | QAOSY18 |
| QAOKP09A | QAOSY10 | |

These files are shipped journaled to QAOSDIAJRN. Should journaling end for any of these files, the STRJRNPf (Start Journal Physical File) command can be used to begin journaling again.

```
STRJRNPf FILE(QUSRSYS/file_name_) JRN(QUSRSYS/QAOSDIAJRN)
          IMAGES(*AFTER) OMTJRNE(*OPNCLO)
```

Certain office functions use commitment control to insure the integrity of the office database files. These files are all journaled to QAOSDIAJRN in library QUSRSYS.

Your application can use the QAOSDIAJRN journal or use its own journal. To avoid potential conflicts between your application and directory services or document library services, your application should use its own journal file.

The office function commits pending database changes before it ends if the function requested completes normally. The office function rolls back pending database changes before it ends if the function does not complete normally. Your application can call an office function without committing or rolling back the changes. The changes will not be committed or rolled back by an office function.

The journal receiver attached to journal QAOSDIAJRN grows as changes are made to the database files listed in Figure 9-8. The journal receiver can get very large if it is not managed properly. Because you are responsible for managing the journal receiver, detach and save the active journal receiver and attach a new one on a regular basis. If you are concerned about storage, do not wait until the receiver is

full. Change the receiver while the journaling function is active. You should not have to end journaling on any of the database files listed in Figure 9-8. The Change Journal (CHGJRN) command should be used to detach an active journal receiver and create a new one. The following example shows how you might use the CHGJRN command:

```
CHGJRN JRN(QUSRSYS/QAOSDIAJRN) JRNRCV(*GEN)
```

This command automatically detaches the current receiver and creates a new receiver. You do not need to know the name of the current receiver. For example, if the current receiver is named QAOSDIAC01, the new receiver is named QAOSDI0001.

Once the journal receiver has been detached and saved, it can be deleted from the system using the Delete Journal Receiver (DLTJRNRCV) command. You can also delete the saved journal receiver by using the Work with Journal Attributes display (WRKJRNA) command and selecting the Journal Receiver Directory display. You can delete journal receivers daily or you can delete them after a complete save.

Incorporate a strategy for managing the journal receiver into your daily cumulative save strategy. See the chapter on database recovery in the *Advanced Backup and Recovery Guide* for more information about journal management and commitment control.

Managing Storage Used by Distribution Objects

The storage used by distribution objects is not charged to individual users. Because these objects are internal system objects and not visible, unnecessary objects can waste storage. This occurs if you do not maintain your outgoing and incoming mail logs. Mail that has been processed or filed should be deleted from the mail logs, reducing the amount of storage used. The section “Maintaining Mail Logs” on page 4-73 gives an example of how the Query Distribution (QRYDST) command and the Delete Distribution (DLTDST) command can be used to help manage these distribution objects.

Office Object Individual Recovery

An AS/400 object is considered damaged when it cannot be read from online storage. The recovery of that object depends on its type. This section discusses how various office objects can be recovered.

Document

If a document is damaged, you can try to recover it using the Reclaim Document Library Object (RCLDLO) command. If this is unsuccessful, you can restore it from the latest saved version using the RSTDLO command. All other operations issued on the damaged document will simply report that it is damaged.

If a document is recovered by restoring it from a saved version, private authorizations to it are lost. The public authority is restored with the document. If the document was marked as personal or non-personal, the marking is restored. If an authorization list is associated with the saved document and that list still exists on the system when the document is restored, the authorization list association also is restored with the document. If private authorities are needed to override public

authority or authorities in the authorization list, they must be established again by the owner.

Folders

If a folder is damaged, reclaim it using the RCLDLO command. Reclaiming a folder constructs it again from the document attributes maintained in database files in the QUSRSYS library. You do not need authority to a folder to reclaim it, nor do you need to be enrolled in the system distribution directory.

Depending on the severity of the damage, the accuracy of the following information may be affected after the folder is reclaimed:

- Authorizations to the folder may be lost and the public authority reset to *EXCLUDE. The owner of the folder, the security officer, or anyone with *ALLOBJ authority can establish explicit authority, authorization list authority, public authority, or access code authority again by using the EDTDLOAUT command.
- The document profile used by the OfficeVision/400 text editor may have been reset to the default system value for a folder accessed by way of the editor. You may change this profile by selecting option 8 (Details) on the OfficeVision/400 Work with Folders display.
- The PC file attributes may be reset to a default value for a folder accessed as a PC directory or subdirectory. Changes to establish the PC file attributes again need to be done from the personal computer.

Internal Document Library Objects

If internal system objects become damaged, all office activity should be stopped until those objects are recovered. A message informs you that a RCLDLO command with the DLO(*INT) option is necessary. The following restrictions apply to reclaiming an internal document library object:

- The RCLDLO DLO(*INT) command can only be run by the security officer or someone with *ALLOBJ or *SECADM special authority.
- While performing this recovery, all office activity using folders or documents must be ended. You may first need to end all subsystems by using the ENDSBS SBS(*ALL) or the ENDSYS command.

Logically Damaged Documents or Folders

A document library object is considered to be logically damaged when erroneous internal OS/400 data areas prevent it from being processed properly. This could result from a programming error or from the system ending abnormally.

Logically damaged objects that were in use when the system ended abnormally are automatically recovered after the system is started again. If you want to use a particular folder or document before the system can recover it, you can reclaim it with the RCLDLO command. You do not need any authority to the folder or document to reclaim it, nor do you need to be enrolled in the system distribution directory. You can also recover logically damaged internal document library system objects by reclaiming the objects as described in "Internal Document Library Objects."

Damaged Database Files

Figure 9-9 shows the physical files shipped with the AS/400 system as part of office services and located in QUSRSYS library. All the files, with the exception of QAOSRH01 and all QAOC* files, are journaled.

Figure 9-9. Physical Files

| | | | | |
|----------|----------|----------|----------|----------|
| QAOSSS10 | QAOSSS18 | QAOKP08A | QAOSRH01 | QAOCIN5 |
| QAOSSS11 | QAOKP01A | QAOKP09A | QAOSAH05 | QAOCIT5 |
| QAOSSS12 | QAOKP02A | QAOKPX4A | QAOSAH07 | QAOCMA5 |
| QAOSSS13 | QAOKP03A | QAOKPCLA | QAOCAU5 | QAOCRDFN |
| QAOSSS14 | QAOKP04A | QAOKPLCA | QAOCCL5 | QAOCXT5 |
| QAOSSS15 | QAOKP05A | QAOKPLGA | QAOCGM5 | |
| QAOSSS17 | QAOKP06A | QAOKPSPA | QAOCGR5 | |

The following files are shipped with the AS/400 system if text search services is installed:

Figure 9-10. Text Search Files

| | | |
|-----------|---------|----------|
| QABBADMTB | QABBDEX | QABBFIX |
| QABBQTB | QABBDIC | QABBLADN |
| QABBCAN | QABBDIX | |
| QABBCOX | QABBDOX | |

The following logical files are also shipped with the AS/400 system as part of office services:

Figure 9-11. Logical Files

| | | | | |
|----------|----------|----------|---------|----------|
| QAOKLABA | QAOKL05A | QAOSAO14 | QAOSI33 | QAOCU5U |
| QAOKLAKA | QAOKL06A | QAOSAO17 | QAOSI34 | QAOCCL5M |
| QAOKLCLA | QAOKL08A | QAOSAO51 | QAOSI35 | QAOCCL5N |
| QAOKLDKA | QAOKL10A | QAOSI12 | QAOSI36 | QAOCCL5O |
| QAOKLEAA | QAOKL11A | QAOSI15 | QAOSI37 | QAOCGM5M |
| QAOKLLGA | QAOKL12A | QAOSI16 | QAOSI38 | QAOCGM5O |
| QAOKLLKA | QAOKL13A | QAOSI18 | QAOSI61 | QAOCGR5N |
| QAOKLLNA | QAOKL27A | QAOSI19 | QAOSI62 | QAOCGR5O |
| QAOKLRKA | QAOKL50A | QAOSI20 | QAOSI63 | QAOCIN5N |
| QAOKLSNA | QAOKL52A | QAOSI21 | QAOSI64 | QAOCIN5P |
| QAOKLSTA | QAOKS01A | QAOSI22 | QAOSI65 | QAOCIN5U |
| QAOKLXDA | QAOKS02A | QAOSI23 | QAOSI66 | QAOCIT5A |
| QAOKLXOA | QAOKS03A | QAOSI25 | QAOSI67 | QAOCIT5M |
| QAOKLXPA | QAOKS04A | QAOSI26 | QAOSI68 | QAOCIT5R |
| QAOKL01A | QAOKS05A | QAOSI29 | QAOSI69 | QAOCIT5S |
| QAOKL02A | QAOSAO09 | QAOSI30 | QAOSI70 | QAOCIT5T |
| QAOKL03A | QAOSAO10 | QAOSI31 | QAOSI71 | QAOCMA5A |
| QAOKL04A | QAOSAO11 | QAOSI32 | QAOSI72 | |

If the damaged file is a document library physical file, QAOSSS*, you must first delete all dependent logical files and the damaged physical file. Then all dependent logical and physical files must be restored. Apply journal changes or run RCLDLO DLO(*ALL) to make these files have required document details for documents and folders on the system. If a physical file other than QAOC* is found to be unusable due to damage or accidental alteration, you can recover it by restoring it from the most recent saved version using the RSTOBJ command, and applying journal changes by using the APYJRNCHG command.

If SAVCHGOBJ used in your daily backup is using the default option of OBJJRN(*NO), the most recent save version is on the complete save backup. If the option used in your daily backup is OBJJRN(*YES), the most recent save version is on your latest daily save. If you choose to delete journal receivers as part of your daily backup, you need to go back to the all daily saves and restore the journal receivers to completely apply journal changes. See the *Basic Backup and Recovery Guide* for more information about applying journal changes.

If one or more of the calendar physical files, QAOC*, are unusable due to damage or accidental alteration, you must restore all of the QAOC* files from the most recent save version using the RSTOBJ command.

If the physical file is damaged, you need to delete it using the DLTF command before you can restore it. If the physical file has logical files associated with it, you need to delete those logical files before deleting the physical file. Use the DSPDBR command to determine the logical files related to the physical file. If you delete any logical file in order to delete and restore a damaged physical file, you must also restore the deleted logical files.

If a logical file is damaged, you can recover it by deleting the file and restoring it from the most recently saved version. Restoring the logical file causes the database access paths to rebuild.

Refer to the *Basic Backup and Recovery Guide* for more information about recovering database files.

Damaged QUSRSYS or QDOC System Library

If you have a damaged library, you may be able to recover it by reclaiming storage using the RCLSTG command or by starting the system again. If not, you will have to restore from a saved version.

If you restore the QUSRSYS library by using the RSTLIB command, the document attributes in the restored database files may not match the documents and folders in the QDOC library. You may need to perform the recovery described in “QDOC and QUSRSYS Library Mismatches.”

It is not necessary to run the RCLDLO DLO(*ALL) command every time library QUSRSYS is restored.

If you attempt to restore QAOSSS* files to library QUSRSYS using the RSTLIB or RSTOBJ command, a check is made to determine whether files QAOSSS10 through QAOSSS18 are already on the system. If a file already exists, then the file is not restored from the save media, and message CPI8A17 is sent for each file.

If files QAOSSS10 through QAOSSS18 do not exist on the system, then the files are restored and a check is done to determine whether QDOC is empty. If QDOC is empty, then the restored files are cleared. If QDOC is not empty, the file is not cleared and message CPD8A71 is sent to inform the user that a RCLDLO DLO(*ALL) command must be run.

A RCLDLO DLO(*ALL) command must be run after a restore of QAOSSS* files to library QUSRSYS if message CPD8A71 was sent for one or more of the QAOSSS* files. If no CPD8A71 messages were sent, then you do not need to run the RCLDLO command.

Notes:

1. Restoring the QUSRSYS library does not restore internal distribution objects such as distribution recipient indexes or distribution tracking objects.
2. DIA files are already on the system and are not restored. For each file you receive a CPI8A17 message.

QDOC and QUSRSYS Library Mismatches

QDOC and QUSRSYS are considered to be mismatched if there is a document or folder object in QDOC that does not have attributes in QUSRSYS, or if there are attributes in QUSRSYS for a document or folder that does not exist in QDOC.

You can recover from mismatched libraries by reclaiming all folders and documents using the RCLDLO DLO(*ALL) command. This updates the database files with document attributes that exactly match all folders and documents as they currently exist in QDOC. Using the *ALL parameter on the RCLDLO command has the same restrictions as the *INT parameter described previously in “Internal Document Library Objects” on page 9-12.

Chapter 10. Miscellaneous Services

This chapter describes control language (CL) commands that are important OfficeVision/400 applications but are not applicable to office services.

Miscellaneous Services Overview

Some of the OfficeVision/400 commands do not belong in a specific category and are described in this section.

- Use the Configure VM/MVS Bridge (CFGRPDS) command to use the VM/MVS bridge application.
- Use the Start Office (STROFC) command to display the OfficeVision/400 menu from a command line.
- Use the Change Office Attributes (CHGOFCA) command to change the contents of a specified OfficeVision/400 value.
- Use the Retrieve Office Attributes (RTVOFCA) to retrieve the current content of specified OfficeVision/400 attributes.
- Use the Delete Outgoing Mail (DLTOUTMAIL) command to delete outgoing mail that has been on the system for a specified number of days.
- Remote document library processing gives you information on how to access documents stored on another attached system.
- Use the Work with Document Library (WRKDOCLIB) command to manage the Document Interchange Architecture (DIA) library services available on remote systems in the network.
- Use the enrollment commands to maintain user enrollment in OfficeVision/400. The following enrollment commands are described:
 - Add Office Enrollment (ADDOFCENR)
 - Change Office Enrollment (CHGOFCENR)
 - Remove Office Enrollment (RMVOFCENR)
 - Retrieve Office Enrollment (RTVOFCENR)

More information about these commands is available in the *CL Reference* manual.

Configure VM/MVS Bridge (CFGRPDS) Command

The Configure VM/MVS Bridge (CFGRPDS) command configures the VM/MVS bridge application. The VM/MVS bridge is an application that provides distribution services between an AS/400 Systems Network Architecture Distribution Service (SNADS) network and a VM/370 network. With the CFGRPDS command, users can configure SNADS and RSCS destinations, and work with the system distribution directory.

The CFGRPDS command has no parameters.

For more information about the VM/MVS bridge, see the *Distribution Services Network Guide*. For more information about the CFGRPDS command, see the *CL Reference* manual.

Start Office (STROFC) Command

The Start Office (STROFC) command allows you to use the office functions provided by OfficeVision/400.

You must be enrolled in the system distribution directory and enrolled in OfficeVision/400 to use this command. You must also have the OfficeVision/400 licensed program installed.

The STROFC command allows you to work with various functions of OfficeVision/400, such as calendars, mail, and library services. For additional detailed information on the STROFC command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

You must be enrolled in OfficeVision/400 and the system distribution directory to use this command. The security officer or an office administrator can enroll users. The security officer must enroll the first user, and can make any user an administrator.

Some Ways to Use This Command

The following are ways you can use the STROFC command:

- When a user is enrolled in OfficeVision/400, all of the functions from that licensed program are available. STROFC can be entered without any parameters to present a menu from which you can select an option, or it can be entered with a parameter to go directly to an option, bypassing the main menu.
- At times the preference is to restrict some or all users to just a few of the office functions. To do this the customer can write an application for producing a menu. When a user signs onto the system, the menu he sees is the one produced by the customer application.

Certain options on the customer menu can be functions from the OfficeVision/400 licensed program. When a user selects one of these options, the customer application can access the office function by using the STROFC command with one parameter that corresponds to the office main menu option number.

When this application is started, the OfficeVision/400 program options made available by the application appear, as do any options that are not office defined by that application.

Rules

The Attention key used by the Office product is not active when STROFC is used with a parameter. However, the user's application can use an attention function with the SETATNPGM and TFRGRPJOB commands.

The STROFC command cannot be used if OfficeVision/400 is already active.

The New mail waiting message is not automatically displayed unless OfficeVision/400 is active. To determine whether mail is waiting for a given user when OfficeVision/400 is not active, customers who write their own applications to display menus and so on may want to investigate using the Query Distribution (QRYDST) command with the Receive Message (RCVMSG) command to period-

ically check a user's distribution status and send a break message to indicate the arrival of incoming distribution.

Change Office Attributes (CHGOFCA) Command

The Change Office Attributes (CHGOFCA) command allows you to change the contents of the specified OfficeVision/400 value.

The CHGOFCA command allows you to select if users not already enrolled in the system distribution directory and in OfficeVision/400 should be automatically enrolled when OfficeVision/400 services are requested. This command also allows you to specify if all users should be allowed use of distribution services commands when the electronic mail services option of the OfficeVision/400 product is not installed.

Security Considerations

You must have *SECADM authority to use the CHGOFCA command.

Rules

You must have the OfficeVision/400 licensed program installed to use this command.

Retrieve Office Attributes (RTVOFCA) Command

The Retrieve Office Attributes (RTVOFCA) command retrieves the current content of the specified OfficeVision/400 attributes. The values are placed into CL variables as specified by the user.

The information returned when the RTVOFCA command is used includes:

- If the user is automatically enrolled in the system distribution directory
- If all users are allowed to use distribution commands when the mail option is not installed
- If calendar services or electronic mail services options are installed
- If the text search option of the OfficeVision/400 product is installed
- If the editor option of the OfficeVision/400 product is installed

Security Considerations

You must have *SECADM authority to use the RTVOFCA command.

Rules

You must have the OfficeVision/400 licensed program installed to use this command.

Delete Outgoing Mail (DLTOUTMAIL) Command

The Delete Outgoing Mail (DLTOUTMAIL) command allows you to delete all outgoing mail that has been on the system for a specified number of days and was sent without confirmation of delivery. For additional detailed information on the DLTOUTMAIL command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

You must be the security officer or have *ALLOBJ authority to use the DLTOUTMAIL command.

Rules

You must have the OfficeVision/400 licensed program installed to use this command.

The DLTOUTMAIL command submits a batch job to delete the outgoing mail.

Note: The job submitted by the DLTOUTMAIL command does not run under adoptive authority.

The DLTOUTMAIL command can be scheduled using the Submit Job (SBMJOB) command.

Remote Document Library Processing

Remote document library processing refers to accessing documents stored on another attached system. The AS/400 system allows users to access documents stored on another remote system, and allows users on another remote system to access documents in the AS/400 document library. An AS/400 system supports access to the following systems:

- System/370 DISOSS
- System/38
- Another AS/400 system

An AS/400 system supports access from the following systems:

- Displaywriter
- Personal Services/PC
- 5520
- System/36
- Another AS/400 system

Accessing a Remote Library from an AS/400 System

The SNADS subsystem must be started before the queued remote document library requests can be processed. The operator has control over these requests by using the SNADS queue management functions, such as:

- Holding and releasing queues
- Holding, releasing, and deleting queue entries

If the job in the SNADS subsystem that is servicing a SNADS *DLS queue should end, the WRKDSTQ command can be used to start the queue sending again.

Remote Document Library System Configuration: Each remote system must be predefined before the corresponding document library can be accessed. A SNADS distribution queue (type *DLS) must be defined for each as well.

Note: Because all remote document library requests are placed on a SNADS queue and processed at a later time, all SNADS queuing support (such as queue depth, time of day sending, or force) can be used for a *DLS queue in the same way a normal SNADS distribution queue can be used.

Follow these guidelines to configure the remote library:

1. Use point-to-point communication objects (line, controller, and device), unless advanced peer-to-peer networking (APPN*) is used for intermediate node routing. **APPN** is data communications support that routes data in a network between advanced program-to-program communications (APPC) systems.
 2. Configure a distribution queue of type *DLS using CFGDSTSRV.
 3. Define the remote library (WRKDOCLIB) using the following:
 - An 8-character remote library name
 - SNADS distribution queue name
 - Remote system type (AS/400 system, System/38, or DISOSS) and the release level
 - Remote library description (must be named and logically attached to the SNADS *DLS queue)
- Note:** Because OfficeVision/400 programs present unique displays for the remote library being used, be certain that the correct support level is defined for each remote library system.
4. Optional: Use keywords, access codes, and document class definitions (OfficeVision/400 administration function needed).

The Work with Document Libraries (WRKDOCLIB) CL command can be used to do the following:

- Add a remote library entry to the network configuration
- Delete a remote library entry from the network configuration
- Change a remote library entry in the network configuration
- Display the remote library entries in the network configuration

Remote System Enrollment/Authorization: Access to the remote system requires that the requesting user be enrolled at the remote system and that the user is authorized to the objects to be accessed on the remote system. User enrollment and authorizations to objects (if any exist) must be done on the remote system. OfficeVision/400 remote library services includes support to authorize users to documents directly (if the remote system is an AS/400 system) or indirectly by the assignment of access codes to documents (DISOSS or System/38). Users working on behalf of another user at the remote system must be enrolled and authorized to work on that user's behalf on that remote system. Remote user identifiers may or may not be the same ones used locally.

Remote Document Library Access: Remote document library access support includes the following capabilities:

- Filing documents into the remote library
- Searching for documents stored in the remote library
- Retrieving documents stored in the remote library
- Changing documents stored in the remote library
- Deleting documents stored in the remote library
- Printing documents stored in the remote library
- Retrieving documents stored in the remote library for change and subsequent replacement in the remote library (AS/400 system only)

These functions are only available through OfficeVision/400. Users must be enrolled in the system distribution directory as well as in OfficeVision/400 to access

remote document libraries. The CL commands used to access the local document library objects, listed in “Host Display Station Interface” on page 5-3, cannot access remote document library objects.

Remote Document Library Request Processing: Remote library requests are not done immediately but are queued for subsequent processing by the internal QDIA job in the QSNADS subsystem. The OfficeVision/400 user may periodically check the status of remote library requests.

Allowing Remote Users to Access an AS/400 System

Remote users must be enrolled in the system distribution directory before these users can access the document library on an AS/400 system. Each user must have a user profile and be authorized to all objects that are accessed. Users working on behalf of another user must be enrolled and authorized to work on that user’s behalf on the AS/400 system. Use the Grant User Permission (GRTUSRPRM) command or the office administration functions to authorize a user to work on another’s behalf. This user identifier may or may not be the same one used at each user’s remote system.

Work with Document Library (WRKDOCLIB) Command

The Work with Document Library (WRKDOCLIB) command allows the user to manage the DIA library services available on remote systems in the network. Remote document library processing refers to accessing documents stored on another attached system.

Security Considerations

You must have security officer or security administrator authority to use this command.

Some Ways to Use This Command

This command lists all the remote libraries and their descriptions. You can add, delete, change library details, and display library details for all remote libraries configured on the system. When you add a remote document library, you can also view the distribution queues that exist and work with them by pressing (F9).

Rules

The following rules apply to the WRKDOCLIB command:

- The remote library requires a name, a queue name, a remote support level, and a description.
- You can only specify a *DLS queue as the name of the SNADS distribution queue.
- This command can only be used interactively.
- AS/400 system supports access to the following systems:
 - System/370 DISOSS
 - System/38
 - Another AS/400 system
- AS/400 system supports access from the following systems:
 - Displaywriter

Personal Services/PC
5520
System/36
Another AS/400 system

Refer to the *CL Reference* manual for the CFGDSTSRV and WRKDSTQ commands, which are related to WRKDOCLIB.

Enrollment Commands

The enrollment commands allow you to maintain user enrollment in OfficeVision/400. The following commands are described in this section:

- Add Office Enrollment (ADDOFCENR)
- Change Office Enrollment (CHGOFCENR)
- Remove Office Enrollment (RMVOFCENR)
- Retrieve Office Enrollment (RTVOFCENR)

Add Office Enrollment (ADDOFCENR) Command

Use the Add Office Enrollment (ADDOFCENR) command to enroll in OfficeVision/400. Before you can be enrolled in OfficeVision/400, you must have a user profile and a system distribution directory entry. You are identified by your user profile. This command has parameters for all items stored in the enrollment record. The parameters for items stored in the user profile and system distribution directory entry are not provided. A calendar cannot be added if the calendar services option is not installed on the system.

To add a new OfficeVision/400 user, you must use the following commands in this order:

1. Create User Profile (CRTUSRPRF)
2. Add Directory Entry (ADDDIRE)
3. Add Office Enrollment (ADDOFCENR)
4. Grant Access Code Authority (GRTACCAUT)
5. Grant User Permission (GRTUSRPMN)

Note: The GRTACCAUT and GRTUSRPMN commands are not required. These commands can be used if you need to add these additional features.

For additional detailed information on the ADDOFCENR command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

The user must have a user profile and a system distribution directory entry before this command can be used to enroll the user in OfficeVision/400. The issuer of this command must be enrolled in OfficeVision/400 as an administrator or have *SECADM authority.

Some Ways to Use This Command

The following examples show the process of creating and defining a user on the system and how to use the ADDOFCENR command to enroll a user in OfficeVision/400.

```
CRTUSRPRF USRPRF(JACKSON) CURLIB(TEMPLIB) LMTCPB(*PARTIAL)
MSGQ(JACKLIB/PROG1)
```

The Create User Profile (CRTUSRPRF) command identifies a user on the system and creates a user profile that contains the values assigned to that user. Jackson is the name assigned to this user profile. Some of the other parameters although are not required, are also given.

```
I ADDDIRE USRID(JACKSON CHICAGO) USRD('JACKSON, Paul W.') USER(JACKSON)
I ADDR1('1160 East Lake Drive') ADDR2('Rochester, MN 55901')
I TELNBR1('507-280-1234')
```

The Add Directory Entry (ADDDIRE) command allows you to add new entries to the system distribution directory. The directory contains information about a user, such as the user ID and address, system name, user profile name, mailing address, and telephone number.

```
I ADDOFCENR USRPRF(JACKSON) WRDPROC(*STANDARD) DFTFLR(JACKSON)
I INLCAL(JACKSON) SCHEDCAL(JACKSON) CALFMT(*WEEKLY)
I CALTIME(08:00 12 30) CALSTRDAY(*MON) CALNBRCOL(5 1)
```

The Add Office Enrollment (ADDOFCENR) command allows you to add a user to OfficeVision/400. The user must already have a user profile and a system distribution directory entry to become enrolled.

```
ADDDSTLE LSTID(JACKSON DISTLIST) USRID((NELSON CHICAGO))
```

The Add Distribution List Entry (ADDDSTLE) command allows you to add new entries to an existing distribution list.

```
GRTACCAUT ACC( 10 15 20 25) USER(JACKSON)
```

The Grant Access Code Authority (GRTACCAUT) command allows you to give the specified users authority to access documents and folders associated with the access codes.

```
GRTUSRPMN TOUSER(JACKSON) FORUSER(NELSON)
```

The Grant User Permission (GRTUSRPMN) command allows you to grant permission to a user to handle documents and folders or to perform related tasks on behalf of another user.

Change Office Enrollment (CHGOFCENR) Command

The Change Office Enrollment (CHGOFCENR) command allows you to change the enrollment of a user who is already enrolled in OfficeVision/400. The user is identified by their user profile. This command has parameters for all items stored in the enrollment record. Parameters for the items stored in the user profile or system distribution directory entry are not provided. A calendar cannot be changed if the calendar services option is not installed on the system.

For additional detailed information on the CHGOFCENR command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

You must be enrolled in OfficeVision/400 as an administrator or have *SECADM authority to issue this command.

Some Ways to Use This Command

The following example shows how to change the enrollment of a user defined on a system and in OfficeVision/400 using the CHGOFCENR command:

```
CHGOFCENR USRPRF(JACKSON) WRDPROC(*STANDARD) DFTFLR(JACKSON)
INLCAL(JACKSON) SCHEDCAL(JACKSON) CALFMT(*WEEKLY)
CALTIME(08:00 12 30) CALSTRDAY(*MON) CALNBRCOL(5 1)
```

Remove Office Enrollment (RMVOFCENR) Command

The Remove Office Enrollment (RMVOFCENR) command removes a user from OfficeVision/400. The user is removed only from OfficeVision/400. The user's system distribution directory entry is not removed, nor is the user's user profile deleted. The user is identified by the user profile.

To remove a user from OfficeVision/400 and the system, you need to use the Remove Office Enrollment (RMVOFCENR) command, the Remove Directory Entry (RMVDIRE) command, and the Delete User Profile (DLTUSRPRF) command.

The *DLT or *NODLT parameter on the RMVOFCENR command indicates that a user can be removed if that user owns objects.

The *CHGOWN operation is not included in the OWNBJOPT parameter. The owned OfficeVision/400 objects of one user cannot have their ownership transferred to another specified OfficeVision/400 user. This can be done with OfficeVision/400 administration menus or through CL commands.

For additional detailed information on the RMVOFCENR command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

You must be enrolled in OfficeVision/400 as an administrator or have *SECADM authority to issue this command.

Some Ways to Use This Command

The Remove Office Enrollment (RMVOFCENR) command removes a user from OfficeVision/400. All owned calendars and personal directories are deleted according to this example.

```
RMVOFCENR USRPRF(JACKSON) OWNBJOPT(*DLT)
```

The Change Document Library Object Owner (CHGDLOOWN) command transfers documents or folder ownership from one user to another user; then the second user becomes the new owner.

```
CHGDLOOWN OWNER(JACKSON) NEWOWN(NELSON)
```

The Remove Directory Entry (RMVDIRE) command allows you to remove a specific user entry from the system distribution directory. In this example, all descriptions for the given user ID and address are removed.

```
RMVDIRE USRID(JACKSON CHICAGO) USRD(*ALL)
```

The Delete User Profile (DLTUSRPRF) command allows you to delete a user profile from the system. The owned objects for the user profile JACKSON are transferred to the specified user profile, JEFF.

```
DLTUSRPRF USRPRF(JACKSON) OWNBJOPT(*CHGOWN JEFF)
```

Retrieve Office Enrollment (RTVOFCENR) Command

The Retrieve Office Enrollment (RTVOFCENR) command allows an authorized user to get information about an OfficeVision/400 user.

The information returned when the RTVOFCENR command is used includes:

- How the user is notified of messages
- The name of the user program called, the library name used to locate the program, and the text displayed when the user selects Option 50 on the OfficeVision/400 menu
- If the user is allowed to use CL commands when working with text documents
- The type of word processing function used
- The name of the first personal directory used when the user runs personal directory functions
- The name of the folder used when a folder is needed but not specified
- User calendar information

For additional detailed information on the RTVOFCENR command, see Appendix D, "Control Language (CL) Commands."

Security Considerations

You must have *USE authority for the user profile for which enrollment information is being retrieved.

Example Applications

This section offers application examples for the STROFC command.

STROFC Examples

The STROFC command has one parameter, OPTION. Its values are *SELECT (the default) and numbers 1 through 9, or 50. If you select one of the numbers as a value for OPTION, the OfficeVision/400 main menu does not appear. Instead, the display corresponding to the menu option number appears. OfficeVision/400 Attention key handling is not active if the Attention key is pressed. However, any Attention key event defined before entering OfficeVision/400 remains active.

From the Command Line

You can use the STROFC command from the command line. For fast access to an OfficeVision/400 function, you can type STROFC # where # is the selection number from the OfficeVision/400 menu. For example, to bring up mail in OfficeVision/400, you would type STROFC 2.

From a CL Program

You can also use STROFC in a CL program. If you do not like to use option numbers you can use words for the options instead. You can create a program that interprets those words into OfficeVision/400 options and builds a command on top of that program.

For example, you can have the command OFC where the source is:

```

/*****/
/*                                          */
/*  COMMAND NAME: OFC                      */
/*                                          */
/*  COMMAND TITLE: STROFC with option     */
/*                                          */
/*  DESCRIPTION: This command brings up STROFC # for a word */
/*                corresponding to the selection number.    */
/*                                          */
/*  CALENDAR - Brings up calendars        */
/*  MAIL - Brings up the mail             */
/*  MESSAGES - Brings up sending messages option */
/*  NOTES - Brings up sending notes option */
/*  DOCS - Work with documents            */
/*  FOLDERS - Work with folders           */
/*  WP - Word processing                  */
/*  DIRDSTL - Directory and Distribution menu */
/*  DIRECTORY - Work with directory (WRKDIR) */
/*  DISTLIST - Work with distribution lists (WRKDSTL) */
/*  DECISION - Decision support in       */
/*  ADMIN - Administration work in       */
/*                                          */
/*                                          */
/*  To create this command, use the CRTCMD command. This is the */
/*  source file for the command.                                          */
/*                                          */
/*****/
OFC:  CMD  PROMPT('STROFC Word Version')
      PARM  KWD(OPTION)  +
           TYPE(*CHAR) LEN(10) RSTD(*YES) MIN(0) MAX(1) +
           VARY(*NO) EXPR(*YES) PASSATR(*NO) FULL(*NO) +
           PROMPT('Selection value') +
           VALUES(CALENDAR MAIL MESSAGES NOTES DOCS FOLDERS +
                  WP DIRDSTL DIRECTORY DISTLIST ADMIN)

```

Figure 10-1. Using STROFC in a CL Program

The program that the command calls is OFCPGM and the source is:

```

/*****
/* This program brings up an OfficeVision/400 function with words  */
/* instead of numbers in the STROFC command.                        */
/*                                                                    */
/* The options for this command are:                                */
/*   CALENDAR - Brings up calendars                               */
/*   MAIL - Brings up the mail                                    */
/*   MESSAGES - Brings up sending messages option                */
/*   NOTES - Brings up sending notes option                      */
/*   DOCS - Work with documents                                  */
/*   FOLDERS - Work with folders                                  */
/*   WP - Word processing                                        */
/*   DIRDSTL - Directory and Distribution menu                    */
/*   DIRECTORY - Work with directory (WRKDIR)                    */
/*   DISTLIST - Work with distribution lists (WRKDSTL)           */
/*   DECISION - Decision support in                              */
/*   ADMIN - Administration work in                              */
/*                                                                    */
/* For example to bring up mail, type on the command line:      */
/* CALL OFCPGM MAIL                                             */
/*                                                                    */
/* Or if the command OFC is created type:                         */
/* OFC MAIL                                                       */
/*****
          PGM          PARM(&OPTION)
          DCL          VAR(&OPTION) TYPE(*CHAR) LEN(10)
          IF (&OPTION *EQ 'CALENDAR') (STROFC 1)
          IF (&OPTION *EQ 'MAIL') (STROFC 2)
          IF (&OPTION *EQ 'MESSAGES') (STROFC 3)
          IF (&OPTION *EQ 'NOTES') (STROFC 4)
          IF (&OPTION *EQ 'DOCS') (STROFC 5)
          IF (&OPTION *EQ 'FOLDERS') (WRKFLR)
          IF (&OPTION *EQ 'WP') (STROFC 6)
          IF (&OPTION *EQ 'DIRDSTL') (STROFC 7)
          IF (&OPTION *EQ 'DIRECTORY') (WRKDIR)
          IF (&OPTION *EQ 'DISTLIST') (WRKDSTL)
          IF (&OPTION *EQ 'DECISION') (STROFC 8)
          IF (&OPTION *EQ 'ADMIN') (STROFC 9)

          ENDPGM

```

Figure 10-2. OFCPGM Program

From a User Menu

The STROFC command can also be used as a command in a user menu. Use the STRSDA command to create the menu. You can restrict your users to access only calendars, mail, and sending messages and notes in office. Using F10 (Work with command) when creating your menu in STRSDA allows you to use the command STROFC 1 to access calendars, STROFC 2 to access mail, STROFC 3 to send messages, and STROFC 4 to send notes. These commands would correspond to the text in the option numbers of the menu that you create.

You can compile your menu when you exit SDA. To access your new menu from the command line, use the command: G0 menuname, where menuname is the name of the user menu you created in SDA.

Initial Program to Call STROFC or Another User Menu

You can also bring up the OfficeVision/400 menu from a sign-on. In OfficeVision/400, use the Administration option (option 9) and select option 1 (Work with enrollment), option 2 (Change a specific user), and option 3 (Change system information). Change the initial program to QOFINLPG and the library to QOFC. This can also be accomplished by using the CHGUSRPRF command INLPGM (Initial Program) for the parameter, and using program QOFINLPG in library QOFC. When that user signs on, the OfficeVision/400 menu is displayed.

This program can be retrieved and the CL source created. Use the RTVCLSRC command to do this. If you want to bring up a user's menu instead of the OfficeVision/400 menu, replace the STROFC statement with a GO menuname statement, where menuname is the name of a user menu. You would compile this CL program and use it as your initial program menu to bring up a user's menu when that user signs on. It should be put into a library other than QOFC to keep the original.

Chapter 11. Enhancing OfficeVision/400

This chapter describes additional functions and products that can enhance your use of OfficeVision/400. The descriptions introduce the products and refer to documentation with more detail.

Operating System Group Job Support (Suspend/Resume)

OfficeVision/400 has a suspend/resume function built into it. The Attention key suspends the current job and returns you to the OfficeVision/400 main menu. Selecting the option of a job that was suspended resumes that job.

This function can be used by a user's application that uses the SETATNPGM and TFRGRPJOB commands.

Examples

The following programs are examples of how to use the suspend/resume function with a menu created from data description specifications (DDS).

The following is the DDS source for the menu used in this program. Use STRSDA, option 1, or enter directly with SEU and CRTDSPF to create the display file.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  
  
A          DSPSIZ(24 80 *DS3)  
A          INDARA  
A          PASSRCD(MENU)  
A          CF12(91 'Return')  
A          R MENU  
A          CHECK(AB)  
A          3 4'Office'  
A          3 11'User'  
A          3 16'Menu'  
A          5 4'1.'  
A          5 7'Work with Documents and Folders'  
A          6 4'2.'  
A          6 7'Mail'  
A          7 4'3.'  
A          7 7'Messages'  
A          8 4'4.'  
A          8 7'Notes'  
A          10 4'Option:'  
A          22 17'F12'  
A          22 21'Return'  
A          OPTION          1 0I 10 13
```

Figure 11-1. DDS Source

The following example is the main program called to bring up the OfficeVision/400 user's menu and start suspend/resume:

```

/*-----*/
/*
/* PROGRAM NAME: MENUPGM
/*
/* PROGRAM TITLE: Office Group Job Example
/*
/* DESCRIPTION: This program is an example of Group Job (Suspend/
/* Resume). It uses options 2, 3, 4 and 5 from the
/* STROFC command and allows the Attention key to
/* suspend and resume the options.
/*
/*-----*/
PGM
DCLF OFCMENU
/*----- S T A R T   O F   E X E C U T A B L E   C O D E -----*/
CHGGRPA GRPJOB(OFCDOC) TEXT('Office Calendar Option')
SETATNPGM PGM(ATTNPGM)
LOOP: SDRCVF
    IF (&IN91 *EQ '1') RETURN /* F12 */
    IF (&OPTION *EQ 1) STROFC 5
    IF (&OPTION *EQ 2) TFRGRPJOB GRPJOB(OFCMAIL) +
        INLGRPPGM(OFCMAILC) +
        TEXT('Office Mail Option')
    IF (&OPTION *EQ 3) TFRGRPJOB GRPJOB(OFCMSG) +
        INLGRPPGM(OFCMSGC) +
        TEXT('Office Messages Option')
    IF (&OPTION *EQ 4) TFRGRPJOB GRPJOB(OFCNOTE) +
        INLGRPPGM(OFCNOTE) +
        TEXT('Office Notes Option')

        GOTO LOOP
ENDPGM

```

Figure 11-2. MENUPGM Program

The following program is called when the Attention key is used:

```

/*-----*/
/*
/* PROGRAM NAME: ATTNPGM
/*
/* PROGRAM TITLE: Attention Key Program
/*
/* DESCRIPTION: This program processes the Attention key from
/*               the group job suspend option for programs
/*               OFCGRPJOB, OFCMMSG, AND OFCNOTEC.
/*
/*-----*/
PGM
DCLF OFCMENU
DCL &GRP *CHAR LEN(10)
/*----- S T A R T   O F   E X E C U T A B L E   C O D E -----*/
CHKRCDLCK
  MONMSG MSGID(CPF321F) EXEC(DO)
    SNDPGMMSG MSG('Attention not allowed. Complete transaction')
    RETURN
  ENDDO
RTVGRPA GRPJOB(&GRP)

SNDRCVF
IF (&IN91 *EQ '1') RETURN /* F12 */
IF (&OPTION *EQ 1) DO
  IF (&GRP *EQ 'OFCDOC') RETURN
  ELSE TFRGRPJOB GRPJOB(OFCDOC)
ENDDO
IF (&OPTION *EQ 2) DO
  IF (&GRP *EQ 'OFCMAIL') RETURN
  ELSE TFRGRPJOB GRPJOB(OFCMAIL) +
    INLGRPPGM(OFCMAILC) +
    TEXT('Office Mail Option')
ENDDO
IF (&OPTION *EQ 3) DO
  IF (&GRP *EQ 'OFCMSG') RETURN
  ELSE TFRGRPJOB GRPJOB(OFCMSG) +
    INLGRPPGM(OFCMSGC) +
    TEXT('Office Messages Option')
ENDDO
IF (&OPTION *EQ 4) DO
  IF (&GRP *EQ 'OFCNOTE') RETURN
  ELSE TFRGRPJOB GRPJOB(OFCNOTE) +
    INLGRPPGM(OFCNOTEC) +
    TEXT('Office Notes Option')
ENDDO
ENDPGM

```

Figure 11-3. ATTNPGM Program

The following programs are called when options 2, 3, or 4 are selected:

```
/*-----*/
/*
/* PROGRAM NAME: OFCMAILC
/*
/* PROGRAM TITLE: Office Mail Program
/*
/* DESCRIPTION: This program calls the attention program if
/*               the Attention key is used. It calls the STROFC 2
/*               option (Mail).
/*
/*-----*/
PGM
/*----- S T A R T   O F   E X E C U T A B L E   C O D E -----*/
SETATNPGM PGM(ATTNPGM)
STROFC 2
ENDPGM

/*-----*/
/*
/* PROGRAM NAME: OFCMSGC
/*
/* PROGRAM TITLE: Office Message Program
/*
/* DESCRIPTION: This program calls the attention program if
/*               the Attention key is used. It calls the STROFC 3
/*               option (Message).
/*
/*-----*/
PGM
/*----- S T A R T   O F   E X E C U T A B L E   C O D E -----*/
SETATNPGM PGM(ATTNPGM)
STROFC 3
ENDPGM

/*-----*/
/*
/* PROGRAM NAME: OFCNOTE
/*
/* PROGRAM TITLE: Office Note Program
/*
/* DESCRIPTION: This program calls the attention program if
/*               the Attention key is used. It calls the STROFC 4
/*               option (Note).
/*
/*-----*/
PGM
/*----- S T A R T   O F   E X E C U T A B L E   C O D E -----*/
SETATNPGM PGM(ATTNPGM)
STROFC 4
ENDPGM
```

Figure 11-4. OFCMAILC, OFCMSGC, and OFCNOTE Programs

See the *Work Management Guide* for information about group jobs and Attention key handling programs.

AS/400 Query

AS/400 Query is an IBM-licensed and decision-support utility used to obtain information from the AS/400 database. You use Query to select, arrange, and analyze information stored in data files to produce reports and other data files. You do not need to understand how to use a high-level programming language to use Query. You can use Query with OfficeVision/400 or DisplayWrite 4 to merge data into word processing documents created by either product.

Query data can be merged into a document in the following ways:

Direct merge

Direct merge queries the database for the information defined in a query.

Column-list merge (OfficeVision/400 only)

Column-list merge takes information from the definition of the query you select. When the document is processed for printing, the query is run and merged into the document in column-list format.

Multicopy merge (OfficeVision/400 only)

Multicopy merge takes information from the definition of the query you select. When the document is processed for printing, multiple customized copies of the document (or multiple letters or labels) are printed.

You can use the MRGDOC or PRTDOC commands to merge the document and query. Refer to “Merge Document (MRGDOC) Command” on page 8-29 or “Print Document (PRTDOC) Command” on page 8-33 for more information about this command. Refer to the *Using OfficeVision/400* Word Processing* manual for information on merging data with shell documents.

See the *Query/400 User's Guide* for more information on AS/400 Query.

AS/400 Business Graphics Utility and Graphical Data Display Manager Command Interfaces

This section briefly describes the **Business Graphics Utility (BGU)** and graphical data display manager (GDDM*) command interfaces to OfficeVision/400.

Business Graphics Utility (BGU)

Using BGU, you can create several types of charts that represent numeric data. You can also create text-only charts. BGU allows you to convert numeric data into:

- Line charts
- Surface charts
- Histograms
- Bar charts
- Pie charts
- Venn diagrams

You can include a graphic in a document using OfficeVision/400 word processing. While editing a document, use the Get Graphic instruction to indicate where you want the graphic printed in your document. This instruction is called by doing the following:

1. Press F5 (goto).
2. Type `.gr` at the prompt.
3. Press the Enter key.

Getting a graphic into a document can only be done interactively.

When the Get Graphic display appears you can do the following:

- View the graphic (by pressing F5)
- Get the graphic (by pressing F14)

You can go to BGU by pressing F14 after doing one of the foregoing functions. The graphic created by BGU must be saved into a graphics data file (GDF) to be used by the text editor. This can be done in BGU by pressing F13.

Follow these two steps to print graphics in your document:

1. Get the graphic into your document using F5 `.gg` or F14 on the Graphic Instruction display.
2. Insert a graphic instruction at all locations where you want the graphic printed.

When the document is processed for printing, the graphic is merged with the document. It can only be viewed separately from the document.

See the *BGU User's Guide and Reference* for detailed information about BGU.

Graphical Data Display Manager (GDDM)

The applications programmer can use GDDM to write programs that take advantage of the power and influence of color graphics. High-level language programs call GDDM routines to construct pictures. High-level languages that support GDDM routines are BASIC, RPG/400, COBOL programming languages, and PL/I. To produce a GDDM application program on the AS/400 system, you must write an AS/400 command in a high-level language.

GDDM can capture the graphic data file (GDF). This file can be saved in a database file. The GDF can be retrieved from the database file and displayed or plotted.

Also, the GDF file can be sent to another system or device for interpretation. System/370, AS/400 system, or System/38 data processing equipment can interpret the GDF.

The GDF can be produced by AS/400 BGU, System/36 BGU, or System/370 families, interpreted by the AS/400 system, and the resulting chart or picture produced on an AS/400 graphics-capable device. The GDF file can be merged with AS/400 documents with the Get Graphic function, using OfficeVision/400 word processing.

Note: Not all GDFs can be brought into documents, not even all those created using AS/400 GDDM. The device token used when calling the DSOPEN routine must not be L79A3. For best results, use a device token of 5292M2 and a blank device name when calling the DSOPEN routine.

See the *GDDM Programming Guide* and the *GDDM Programming Reference* manual for more information.

PC Support/400

PC Support/400 allows you to access the AS/400 system with a personal computer through the work station function. PC Support/400 uses folders to store information such as text documents, programs, files, and documents. The information in a folder can be used by both AS/400 and PC users. A folder can be shared by more than one AS/400 or PC user at a time.

You can access PC Support/400 and data processing applications through the PC Support/400 organizer. The PC Support/400 organizer is available as part of the work station function. The options on the PC Support/400 organizer menu can be changed to run any AS/400 or PC application you use frequently. You can also use your own menu.

While using your personal computer as an AS/400 work station, you can copy data from an AS/400 database file to a PC document (using CPYTOPCD) or from a PC document to an AS/400 database file (using CPYFRMPCD).

PC Support/400 allows you to use printers attached to the AS/400 system as though they were directly attached to your personal computers.

PC Support/400 also allows you to communicate with other AS/400 users by sending and receiving messages. You can communicate with other display stations, other personal computers attached to the AS/400 systems, and other users elsewhere when you are on a network. You can transfer data from the AS/400 system to the personal computer, and from the personal computer to the AS/400 system.

See the *PC Support/400 User's Guide for DOS* or *PC Support/400 User's Guide for OS/2* for detailed information about PC Support/400.

Normal Work Station, Database, Communications, and High-Level Language Support

In your normal line of business, you may use database, communications, normal work stations, and high-level language applications. You can add the office commands by considering the information in this section.

Database Files

Database files are the basis of the AS/400 system and are used throughout PC Support/400. The layout of office database files is described in Appendix C, "Office Services Output File Considerations." CL commands that create and use database files are described in the *CL Reference* manual. See the *Database Guide* for information on how to use database files on the AS/400 system.

Communications

The AS/400 user can communicate with users on the same system (local users) or with users located on a different system (remote users). Using applications, files (including spooled files and save files), documents, messages, and input streams can be distributed to local and remote users.

System Network Architecture Distribution Services (SNADS) can distribute documents, files, messages, and input streams to other systems directly and indirectly attached to your system.

The system distribution directory identifies the name (user ID and address) and system of the authorized users in a SNADS network.

The **VM/MVS bridge** allows AS/400 and VM/370 users to exchange distributions, including OfficeVision/VM documents and notes, data files, and print files.

See the *Distribution Services Network Guide* for more information.

High-Level Language Support

The AS/400 system supports the following high-level languages:

- BASIC
- RPG/400 language
- COBOL/400* language
- PL/I
- PASCAL
- REXX
- C/400* language

These languages and CL are used for application programming. From the high-level languages, programs can access office commands through calls to the CL commands. To do this, call the QCMDEXC program with two parameters. The first parameter is the command (in apostrophes), and the second parameter is the length of the command.

For example, to display the system distribution directory to an output file from a high-level language program, you would type the following:

```
CALL PGM(QCMDEXC)
      PARM('DSPDIR OUTPUT(*OUTFILE) OUTFILE(MYLIB/DIRLIST)' 46)
```

See the *CL Programmer's Guide* for information about the QCMDEXC program.

OfficeVision/400 V2R2 Application Enabler Support

The Version 2 Release 2 support level for the OfficeVision/400 application enabler allows other office applications, such as editors and spreadsheets, to be used from the document handling interfaces of OfficeVision/400. The V2R2 OfficeVision/400 application enabler support can be used with OfficeVision/400 Version 2 Release 3. The purpose of the OfficeVision/400 application enabler is to provide a seamless integration of other office applications into the standard OfficeVision/400 display and command interfaces. The integration of the OfficeVision/400 application enabler may be less than perfect depending on the installation and applications used. The application's characteristics may not be relevant for certain options that

the OfficeVision/400 application enabler supports. For example, the OfficeVision/400 application enabler supports a spell check option that may not be applicable for some types of objects such as spreadsheets. In this case, you may receive an error message indicating that the option is not supported for that type of document. This is a function of the office application or a user exit program.

Using Application Enabler Support

The integration in V2R2 support is accomplished by providing:

- An application program interface (API) or exit from the OfficeVision/400 editor, print, and mail routines. An exit must be provided to call the other office applications from these OfficeVision/400 interfaces to process document requests.
- An API or exit from the OfficeVision/400 document open function routines. This interface defines document conversions to allow OfficeVision/400 to access documents created by other office applications.

When OfficeVision/400 is accessing a document that is not RFTAS400, FFTAS400, RFTDCA, or FFTDCA, it checks whether there are any document conversions before deciding that it cannot process the document.

If no other office applications or document conversions have been defined, OfficeVision/400 processes all requests and uses only the IBM-supplied conversion programs.

If the OfficeVision/400 application enabler V2R2 support is used, processing proceeds as follows:

1. A document request is made using an OfficeVision/400 command or display.
2. If a document handling exit has been registered, the exit program is called with information about the request (such as, function, document name, document type, and so on).
3. The exit program determines if it should call the other office application to process the request or return the request to OfficeVision/400 for processing. If the request is processed by the other office application, you return to the display where the request was made (for example, Work with Documents, Work with Mail).
4. If the request is returned by the document handling exit for OfficeVision/400 to process, OfficeVision/400 handles the request.
5. When OfficeVision/400 opens a document to begin processing, the system requests that document conversions be used, if they are available. The document conversions are used to convert documents that are not AS400 or Document Content Architecture types.
 - If a conversion exit has been registered, the exit is called. The conversion exit returns either the converted document or a message that the requested conversion is not supported.
 - If no conversion exit is registered, OfficeVision/400 checks for document conversions defined using the Add Document Conversion (ADDDOCCVN) and Work with Document Conversions (WRKDOCCVN) commands.
 - If the necessary document conversion has been defined, it is called to perform the conversion.

- When the conversion is complete, the document is returned to OfficeVision/400 for processing in AS400 or Document Content Architecture format.

See Appendix D, “Control Language (CL) Commands” for complete information about the OfficeVision/400 application enabler CL commands.

Setting up Other Office Applications

To set up another office application using V2R2 support, do the following:

1. Create or install a document handling exit program.

The exit that processes the document can be supplied with the office application or can be written for your system. The *System Programmer's Interface Reference* manual contains information on writing an exit program to access another office application. Prototype data structures for the exit program are provided for RPG, COBOL, and C in QUSRTOOL. These structures define the function specific parameters in the document handling exit interface.

2. Define the document handling exit to OfficeVision/400.

The exit must be defined to OfficeVision/400 using the Change Office Exit Programs API, QOGCHGOE. The *System Programmer's Interface Reference* manual provides complete information about using the QOGCHGOE API.

3. Create or install the document conversion exit or document conversion programs.

There are two ways to use the APIs to access document conversions from OfficeVision/400:

- Creating or installing the document conversion exit program to call the document conversions
- Using OfficeVision/400 to call the document conversions directly

You should use the document conversion exit program if:

- You are using another office application that requires one
- You need to be able to completely control the conversion process

In most cases, it is easier to let OfficeVision/400 call the document conversions.

Note: The same parameters are used by OfficeVision/400 to call the document conversions and the conversion exit.

4. Define the document conversion exit or document conversion programs to OfficeVision/400.

The document conversion exit must be defined to OfficeVision/400 using the QOGCHGOE API.

If you are using OfficeVision/400 to manage the conversion process, you can define them to OfficeVision/400 using the ADDDOCCVN and WRKDOCCVN commands. See Appendix D, “Control Language (CL) Commands” for a complete description of these commands.

Note: The Retrieve Office Exit Programs API, QOGRTVOE, can be used to verify the current exit program values. The *System Programmer's Interface Reference* manual provides more information about using the QOGRTVOE API.

Security Considerations

You must be an office administrator to use the document conversion commands and the QOGCHGOE and QOGRTVOE APIs.

OfficeVision/400 affinity (on behalf of) support remains active when other office applications and document conversions are called.

Processing Considerations

You should be aware of the following:

- Exit programs registered using the QOGCHGOE API and document conversions defined using the ADDDOCCVN and WRKDOCCVN commands are used for all OfficeVision/400 users on the system. See “Performance Considerations” on page 11-12 for more information on the effects this might have.
- Documents that are not filed in a folder (for example, mail items and documents that are not in a folder) are temporarily copied to a folder before calling the document handling exit. The temporary copies are deleted when the request is complete.
- PC-based document conversions and applications are also supported using the Start PC Command (STRPCCMD) command in the document handling exit and document conversions. PC organizer must be active for the STRPCCMD command to be successful. The STRPCCMD command is only supported interactively, which means that batch jobs (such as printing on the job queue) are not supported using this technique.
- The document handling exit support is not mutually exclusive with the PC organizer editor of choice function. The editor of choice is determined using the following priority:
 1. Document handling exit
 2. PC organizer profile editor of choice
 3. OfficeVision/400 editor
- Since closed documents are passed to the document handling exit and document conversions, sharing problems can occur. This could result in the document not being available for processing by the other office application or document conversion program.
- Printing mail for indirect users is always processed by OfficeVision/400 print services. However, mail services requests document conversions when opening documents for printing.
- When OfficeVision/400 mail services sends documents, AS400 and Document Content Architecture documents are sent as Document Content Architecture. All other types are sent as is. This means that it may be necessary to make the document conversions available to all systems to which the documents are sent.
- Only Document Content Architecture and PCFILE types are supported by the VM/MVS bridge.
- The other office applications are not called for single record merge. OfficeVision/400 always processes this type of merge request. This type of function can be simulated by creating a one-record file and doing a normal data and text merge from the file.

- The F20 key is shown on the Create Document Details and Change Document Details displays only if there is no document handling exit specified.
- The OfficeVision/400 text profile interfaces do not call the document handling exit. However, you are allowed to activate a document as a text profile even if it is not an AS400 or Document Content Architecture type. To create this text profile, you can copy a document to QPRFFLR or use another office application to create a document in QPRFFLR.

Performance Considerations

Most of the performance impact of the OfficeVision/400 application enabler depends on the document handling exit, the other office applications, and the document conversions.

- Users that do not register a document handling exit program should not experience any degradation.
- Users that register a document handling exit program incur a performance impact even if the program passes the request back to OfficeVision/400 for processing. This is due to the setup, call, and cleanup processing of the document handling exit program.
- The other office applications are started and ended with each function request. For some applications (particularly PC applications that take considerable time), this may result in unacceptable performance.
- Once started, the operational performance of the other office applications using this support should be equivalent to their stand-alone performance.

OfficeVision/400 V2R3 Application Enabler Support

The Version 2 Release 3 support level for the OfficeVision/400 application enabler allows other office applications, such as editors and spreadsheets, to be used from the document handling interfaces of OfficeVision/400. The V2R3 implementation is an enhancement to the set of exit interfaces created in V2R2. A user must be enrolled in OfficeVision/400 before using the V2R3 OfficeVision/400 application enabler support.

OfficeVision/400 users of the OfficeVision/400 application enabler V2R3 support level can choose multiple editors, can define document types, and can select document conversion options. In the V2R2 support level there is one user exit program, any choices for the user need to be in the user code, and only IBM-supplied types can be used.

The OfficeVision/400 application enabler has been enhanced to make customizing your office environment easier. The V2R3 OfficeVision/400 application enabler support assists the user to integrate other office applications into the standard OfficeVision/400. This is accomplished by creating command interfaces that define the applications to OfficeVision/400 in terms of document types. Applications created this way can support the following functions:

- CREATE
- EDIT
- VIEW
- MAILEDIT
- MAILVIEW

- MAILFWD
- MAILREPLY
- PAGINATE
- PRINT
- PRINTOPTS
- MERGE
- MERGEOPTS
- SPELLCHECK
- FILLFORM

Using Application Enabler Support

The integration of the V2R3 OfficeVision/400 application enabler is done by using four new functions:

Document type sets

Document sets are logical groupings of document type definitions.

Document types

Document type definitions define non-DIA user document types.

Office application definitions

Office application definitions describe applications that can be called by OfficeVision/400.

Office application entries

Office application entries define for the user which application to call based on the document set.

Several commands were added to assist the user to work with the OfficeVision/400 application enabler. The following CL commands were added in the V2R3 support level:

- ADDOFCAPPE Add Office Application Entry
- CHGDOCSET Change Document Set
- CHGDOCTYP Change Document Type
- CHGOFCAPPD Change Office Application Description
- CHGOFCAPPE Change Office Application Entry
- CRTDOCSET Create Document Set
- CRTDOCTYP Create Document Type
- CRTOFCAPPD Create Office Application Description
- CVTDOC Convert Document
- DLTDOCSET Delete Document Set
- DLTDCTYP Delete Document Type
- DLTOFCAPPD Delete Office Application Description
- RMVOFCAPPE Remove Office Application Entry
- WRKDOCSET Work with Document Sets
- WRKDOCTYP Work with Document Types
- WRKOFCAPPD Work with Office Application Descriptions
- WRKOFCAPPE Work with Office Application Entries

See Appendix D, "Control Language (CL) Commands" for complete information about the OfficeVision/400 CL commands.

If the OfficeVision/400 application enabler has been activated, the application called will be based on the document type and the active user profile. The OfficeVision/400 application enabler processing follows:

1. A requested action against a document type is made by a user through an OfficeVision/400 command or display.
2. The document set is determined by looking up the document type in the set table.
3. After the document set and user profile name have been identified, the application is determined. The application entry table is searched for the correct application name.
4. The application name is now used to look up the exit handling program that can do the requested function.
5. Lastly, the document is converted to the accepted document type defined in the application description and the exit handling program is called. Document conversions are defined using the ADDDOCCVN and WRKDOCCVN commands.

Setting up Other Office Applications

To set up another office application with V2R3 support, the user can do the following:

1. Create any new document sets supported by the application with the CRTDOCSET command.
2. Add any new document type definitions to the system and identify the set that the new type belongs in with the CRTDOCTYP command.
3. If you have a document set already defined for this type of application, you may need to change the set defined in the existing document types. Use the CHGDOCTYP command to change the set definition.
4. Create or install a document handling exit program.
5. Register the application to the system with the CRTOFCAPPD command. This identifies a new application to the system and identifies to the system the exit program used by the application.
6. Create the office application entry with the ADDOFCAPPE command. This allows the application description to be associated to the user profile.
7. Use the QOGCHGOE API to activate the application. See the *System Programmer's Interface Reference* manual for more information on the QOGCHGOE API.

Note: This is only needed for the first application installed on the system.

8. Create or install either the document conversion exit or any document conversion programs.

There are two ways to use the APIs to access document conversions from OfficeVision/400:

- Creating or installing the document conversion exit program to manage the conversion
- Using OfficeVision/400 to call the document conversions directly

You should use the document conversion exit program if:

- You are using another office application that supplies one
- You need to be able to control the calling of document conversions

In most cases, it is easier to let OfficeVision/400 call the document conversions.

Note: The same parameters are used by OfficeVision/400 to call the document conversions and the conversion exit. See the *System Programmer's Interface Reference* manual for more information on the document conversion exit program.

9. Define the document conversion exit or document conversion programs to OfficeVision/400.

The document conversion exit must be defined to OfficeVision/400 using the QOGCHGOE API.

If you are using OfficeVision/400 to call the document conversion programs, you can define them to OfficeVision/400 using the ADDDOCCVN and WRKDOCCVN commands. See Appendix D, "Control Language (CL) Commands" for a complete description of these commands.

Note: The Retrieve Office Exit Programs API, QOGRTVOE, can be used to verify the current exit program values. The *System Programmer's Interface Reference* manual provides more information about using the QOGRTVOE API.

Security Considerations

You must be an office administrator to use the document conversion commands and the QOGCHGOE and QOGRTVOE APIs.

OfficeVision/400 affinity (on behalf of) support remains active when other office applications and document conversions are called.

Processing Considerations

You should be aware of the following:

- Documents that are not filed in a folder (for example, mail items and documents that are not in a folder) are temporarily copied to a folder before calling the document handling exit. The temporary copies are deleted when the request is complete.
- PC-based document conversions and applications are also supported using the Start PC Command (STRPCCMD) command in the document handling exit and document conversions. PC organizer must be active for the STRPCCMD command to be successful. The STRPCCMD command is only supported interactively, which means that batch jobs (such as printing on the job queue) are not supported using this technique.
- The document handling exit support is not mutually exclusive with the PC organizer editor of choice function. The editor of choice is determined using the following priority:
 1. Document handling exit
 2. PC organizer profile editor of choice
 3. OfficeVision/400 editor
- Since closed documents are passed to the document handling exit and document conversions, sharing problems can occur. This could result in the document not being available for processing by the other office application or document conversion program.
- Printing mail for indirect users is always processed by OfficeVision/400 print services. However, mail services requests document conversions when opening documents for printing.

- When OfficeVision/400 mail services sends documents, AS400 and Document Content Architecture documents are sent as Document Content Architecture. All other types are sent as is. This means that it may be necessary to make the document conversions available to all systems to which the documents are sent.
- Only Document Content Architecture and PCFILE types are supported by the VM/MVS bridge.
- The other office applications are not called for single record merge. OfficeVision/400 always processes this type of merge request. This type of function can be simulated by creating a one-record file and doing a normal data and text merge from the file.
- The F20 key is shown on the Create Document Details and Change Document Details displays only if there is no document handling exit specified.
- The OfficeVision/400 text profile interfaces do not call the document handling exit. However, you are allowed to activate a document as a text profile even if it is not an AS400 or Document Content Architecture type. To create this text profile, you can copy a document to QPRFFLR or use another office application to create a document in QPRFFLR.
- To improve performance, the OfficeVision/400 application enabler maintains a cache containing the program association with a type for each job. Only the first reference to a type needs to run the entire resolution procedure. Any changes to the configuration, such as changing the application entries or set membership for a type, will not appear until the user signs off and signs on. The user should stop all OfficeVision/400 activity when making global configuration changes. If personal application entries have been changed by the user, then the user should sign off and sign on.
- Care must be taken when creating a new document with CRTDOC if the type of the document does not support an editor. The created document will be empty and no editor is called during the CRTDOC session. If an edit request is later made against that document, OfficeVision/400 will convert it to an RFTAS400 document. If a document type supports an editor, then this problem does not occur.
- If a document has a type that is not registered with the system, OfficeVision/400 will attempt to process the document. This association is not maintained in the cache. Any future attempts to edit a document of this same type will result in another attempt by OfficeVision/400 to resolve the application.
- If a document type is a member of *NOSET or there is no application entry, using either *SYSTEM or user profile name, for a set, then OfficeVision/400 will attempt to process the document. This association is maintained in the cache.

Performance Considerations

Most of the performance impact of the OfficeVision/400 application enabler depends on the document handling exit, the other office applications, and the document conversions.

- Generally, users that do not activate the OfficeVision/400 application enabler will not experience any performance degradation.
- Users that activate the OfficeVision/400 application enabler incur a performance impact even if the program passes the request back to OfficeVision/400 for pro-

cessing. This is due to the setup, call, and cleanup processing of the document handling exit program.

- The other office applications are started and ended with each function request. For some applications (particularly PC applications that take considerable time), this may result in unacceptable performance.
- Once started, the operational performance of the other office applications using this support should be equivalent to their stand-alone performance.
- Because the OfficeVision/400 application enabler uses a cache, the first reference to a particular type by a job can take significantly longer than succeeding references. This is due to the resolution process used to determine which application to perform.
- Performing editing functions on documents of type PCFILE can experience significant performance penalties. This is because the PCFILE identification process must be run against them. See "PCFILE Identification Process API (QO1PCFID)" for information on the PCFILE API. If the document cannot be associated with a type, the process will have to run again every time the document is edited. Documents which can be edited should not have a type of PCFILE if the user has the PCFILE identification indicator set to activate status.
- The PCFILE identification process checks types in type number order. The earlier the type is identified, the sooner the process will stop. The user should assign frequently used types lower numbers than types used less frequently.

PCFILE Identification Process API (QO1PCFID)

The PCFILE Identification Process application program interface (API) QO1PCFID will determine the correct file type or a document type of PCFILE. QO1PCFID is passed a document identified by name and an action code. The API will check the document to identify what type it should be and will return the new type. If the action is to change the type, the API will also change the document type.

The QO1PCFID API has these parameters:

| QO1PCFID: Required Parameters | | | |
|-------------------------------|--|--------------|-----------------|
| 1 | Type of the document name | INPUT | CHAR(1) |
| | <p>The type of document name being provided to the API. The following types are supported:</p> <p>0 The document is identified as a document name and the folder containing the document. Both the name and the folder parameters contain meaningful information.</p> <p>1 The document is identified by a system object name. Only the name parameter is meaningful. The folder parameter will be ignored.</p> <p>2 The document is identified by a library assigned document name. This is an internal OfficeVision/400 name for a document. Only the name field parameter contains meaningful information.</p> | | |
| 2 | Document name | INPUT | CHAR(12) |
| | <p>This parameter contains the name of the document to be identified. The exact meaning of the name is dependent on the type of document name parameter.</p> | | |

| QO1PCFID: Required Parameters | | | |
|--------------------------------------|---|---------------|------------------|
| 3 | Folder | INPUT | CHAR(63) |
| | The folder path containing the document to be identified. This parameter is meaningful only if the type of document name parameter is set to 0. | | |
| 4 | Type change indicator | INPUT | CHAR(1) |
| | <p>The indicator which identifies whether the API is to change the type of the document or only return the proposed type. The following values can be used:</p> <p>0 Do not change the type of the document. The user will need *USE authority to the document.</p> <p>1 Change the type of the document. The user will need *CHANGE authority to the document.</p> | | |
| 5 | Document type checked | INPUT | BINARY(4) |
| | <p>This is the document type that OfficeVision/400 checks to verify that PCFILE is a member of. The programmer can use this parameter to write a single PCFILE identification program which will attempt to identify several types. The following values can be used:</p> <p>0 Check all types</p> <p>> 65535 Check fails</p> | | |
| 6 | Methods to use | INPUT | CHAR(5) |
| | <p>This is a flag parameter indicating which identification methods are active. A value of 1 in the relevant byte position indicates the identification method is used. A value of 0, or any value other than 1, indicates that the method is not used.</p> <p>1 Use the extension method with the document name extension. If the type of document name is not folder and document, the name will be determined. This method is ignored if the document is a distribution. Performance benefits can be gained if the document is a distribution and this byte is turned off.</p> <p>2 Use the extension method with the extension on the name in the PCFILE subprofile, if such a value exists.</p> <p>3 Use the folder method of identification. This method is ignored if the document is not filed in a folder. Performance benefits can be gained if the document is not filed in a folder and this byte is turned off.</p> <p>4 Use the value method of identification.</p> <p>5 Use the program method of identification.</p> | | |
| 7 | New type | OUTPUT | BINARY(4) |
| | The proposed new type of the document that is a result of the identification process. If a change was requested, this will be the new type of the file. If no new type could be identified, this field will return a value of 14, or type PCFILE. If an error occurs, the value in this field is undefined. | | |
| 8 | Error Code | I/O | CHAR(*) |
| | For complete information on the error code parameter, see the <i>System Programmer's Interface Reference</i> manual. | | |

PCFILE User Program Interface (PCFI0000)

The PCFILE user program interface (PCFI0000) will be used by a user exit program to check a document for PCFILE identification. For more information on creating document types see "CRTDOCTYP (Create Document Type) Command" on page D-51.

The PCFI0000 interface has these parameters:

| PCFI0000: Required Parameters | | | |
|-------------------------------|---|---------------|------------------|
| 1 | Document name | INPUT | CHAR(12) |
| | This is the name of the document to be checked. This may not be the original name of the document. | | |
| 2 | Folder name | INPUT | CHAR(63) |
| | This is the folder in which the document is found. This may not be the residing folder of the original document. | | |
| 3 | Requested type | INPUT | BINARY(4) |
| | This is the document interchange architecture (DIA) type that the exit program identifies with the document. | | |
| 4 | Success indicator | OUTPUT | CHAR(1) |
| | This indicates if the document being checked is the type which was requested. Note: Any exception issued by the user exit program will be reissued as a diagnostic and will cause the call to be handled as if it had returned a 0 indication. 0 No, this document is not to be treated as the requested type. 1 Yes, this document is to be treated as the requested type. | | |

Appendix A. Character Set and Code Page Considerations

On the AS/400 system, you can specify how your data is represented using character set and code page:

- Character set** A set of graphic symbols treated as an entity (for example, a keyboard or a print wheel on a printer).
- Code page** A set of graphic characters represented by a bit-pattern. Within a code page, each bit-pattern has only one meaning. For example, the # character is represented by hex 7B in code page 037 (USA/Canada). In code page 283 (Spain), the # character is represented by hex 69 and hex 7B represents a tilde (~).

In other words, a code page specifies how each graphic symbol is represented by bits and the character set tells which of those symbols in the code page can be used.

Character sets and code pages are often referred to with the following notation:

xxx yyy

xxx is the character set and yyy is the code page. For example, 697 500 is character set 697 and code page 500.

See the *National Language Support Planning Guide* for more information.

Common Character Sets and Code Pages

The AS/400 system must store some data in common code pages. The directory stores the user ID and address in character set 930 and code page 500 (uppercase international). When a directory entry is added, the user ID and address entered are translated from the character set and code page specified to the character set 930 and code page 500. The translation to common character set and code page is done so the user ID and address can be recognized by other systems with different character sets and code pages.

OfficeVision/400 also uses the following character sets and code pages:

- 697 500** Uppercase and lowercase international
337 256 Uppercase and lowercase international used by DIA

CMDCHRID and CHRID Parameter

Many of the office control language (CL) commands use the CMDCHRID or CHRID parameters. The keyword used on the office CL commands is CMDCHRID. Other commands use the keyword CHRID to specify the character set and code page. CHRID informs the called command as to which character set and code page are used. Not all parameters of the command are affected. See the description of each command to determine which parameters use the character set and code page values.

The character set and code page values tell the command to use the value to translate to a common character set and code page (697/500 or 930/500 or 337/256) or store the character set and code page with the data. For example, for

the QRYDST command, if you specify CMDCHRID (288 297), French, the command translates the graphics characters in the USRID parameter to character set 930 and code page 500.

When the default value *SYSVAL is used, the character set and code page are retrieved from the system values. To change the system's character set and code page, use the following command:

```
CHGSYSVAL SYSVAL(QCHRID) VALUE(character-set-value code-page-value)
```

To display the system's character set and code page value, use the following command:

```
DSPSYSVAL SYSVAL(QCHRID)
```

When you specify *DEV, the character set and code page values come from the display device description. This option is valid only when entered from an interactive job. Use the Work with Device Description (WRKDEV) command to work with the character set and code page values for a device.

Translating Data

AS/400 users can translate data. Translation tables are shipped with the base code. The tables are in libraries QSYS and QUSRSYS. To see the list of translation tables provided, use the following commands:

```
DSPOBJD OBJ(QSYS/*ALL) OBJTYPE(*TBL)
DSPOBJD OBJ(QUSRSYS/*ALL) OBJTYPE(*TBL)
```

When data is sent to an output file by an office CL, sometimes a field in the output file defines what character set and code page the data is in. Your application can translate that data into the character set and code page you want by using the tables supplied with the system or tables you created.

Call the QDCXLATE program to translate the data. For example, in a CL program you could specify the following:

```
CALL PGM(QDCXLATE) PARM(&FLDLEN &FIELD &TBL &LIB)
```

In the example:

- &FLDLEN** The length of the field to be translated. It is a five-digit, packed, decimal field with no decimal positions.
- &FIELD** A character field containing the data to be translated. The data is replaced with the translated data.
- &TBL** A 10-character field containing the name of the translation table to be used. The table name must be left-adjusted.
- &LIB** A 10-character field containing the name of the library that contains the translation table. The library name must be left-adjusted. If this field is not specified, the library list is used to find the translation table.

See the *CL Programmer's Guide* for more information.

For example, to translate the data field (&RCVDTA) in a RCVDST output file using the system model output file layout (QSYS/QAOSIRCV) from character set 101 (in field &RCVCRS) and code page 037 (in field &RCVCPG) to character set 337 and code page 256, you would code the following in CL:

```
CALL PGM(QDCXLATE) PARM(&RCVDL &RCVDTA Q037337256 QUSRSYS)
```

You could then use the FILDOC command to file the document into a folder with the new translated fields. If you do this, update the character set field (&RCVCRS) and code page field (&RCVCPG) to the new translated character set and code page.

Word Processing Character Set and Code Page Considerations

OfficeVision/400 word processing processes character sets and code pages differently than does DisplayWrite/36 when printing. DisplayWrite/36 always prints documents using code page 256 (unless overridden by the print options).

OfficeVision/400 word processing uses a hierarchy to decide what character set and code page to use when printing.

Note: When sending a document to another user or system, 337/256 is normally used for OfficeVision/400 and DisplayWrite/36.

The hierarchy used is as follows:

- *Graphic character set* prompt on page 2 of the Print Options display
- CHRID value on the Override Print File (OVRPRTF) command
- CHRID value in the print file
- QCHRID system value

Print Options Display

The *Graphic character set* prompt on page 2 of the Print Options display takes precedence over anything else.

If you are using a supported code page, but need to print to a 5219 printer using a character set in code page 256, specify the character set ID in the *Graphic character set* prompt. For example, to print a Spanish document (edited using code page 284) to a 5219 printer, use 045 in the prompt. The document translates to code page 256 and prints using 045 256.

If you are using an unsupported code page and need to print to an RPQ printer, specify the character set and code page ID in this prompt. For example, to print a Turkish document (code page 905) use 965 905 at the prompt. The document will not translate (it is assumed that the text was entered using a Turkish keyboard), but it will print using 965 905.

If this method is to be used extensively, set up the print options in a text profile so that newly created documents have the *Graphic character set* prompt set to the values you need.

Override Print File OVRPRTF Command

The CHRID value in the Override Print File (OVRPRTF) command takes precedence next (that is, if the *Graphic character set* prompt on the Print Options display is not used).

If you are using a supported code page, but you must print to a 5219 Printer using a character set in code page 256, you can use the following command:

```
OVRPRTF FILE(file name) TOFILE(library name/print file name) ...  
CHRID(045 256)
```

The command overrides the specified print file for the current session only. Documents you print that use the specified print file translate to code page 256 and print using 045 256 (assuming they are not using the *Graphic character set* prompt on the Print Options display).

Note: If *SYSVAL or *DEV D are used, the CHRID system value is used.

If this method is to be used extensively, set up the OVRPRTF command in an initial program so it is set for each session.

Print Files

The CHRID value in the print file takes precedence next (that is, if the *Graphic character set* prompt on the Print Options display and the OVRPRTF command are not used).

If you are using a supported code page, but need to print to a 5219 Printer using a character set in code page 256, you can create a new print file using the following command:

```
CRTPRTF FILE(library name/print file name) ... CHRID(045 256) ...
```

Or, you change a print file using the following command:

```
CHGPRTF FILE(library name/print file name) ... CHRID(045 256) ...
```

The specified print file can be used by a user and works each time the user signs on without entering another command. Documents you print that use the specified print file (using the *Printer file* prompt on page 1 of the Print Options display) translate to code page 256 and print using 045 256 (assuming they are not using the *Graphic character set* prompt on the Print Options display).

Note: If *SYSVAL or *DEV D are used, the CHRID system value is used.

If a new print file is to be used extensively, set up the print options to use this new print file in a text profile so newly created documents have the *Printer file* prompt set to the values you need.

QCHRID System Value

The QCHRID system value takes precedence last (if the *Graphic character set* prompt on the Print Options display and the OVRPRTF command are not used, and the print file has *SYSVAL or *DEV D).

If you are using a supported code page and you want all your printing to use character set and code page 256, you can change the system value using the following command:

CHGSYSVAL SYSVAL(QCHRID) VALUE(045 256)

Printed documents that use a print file with *SYSVAL or *DEVD translate to the system code page and print using the CHRID system value (assuming they are not using the *Graphic character set* prompt on the Print Options display).

Code Pages Supported by Word Processing

The following code pages are supported by OfficeVision/400 word processing:

| | |
|------------|-----------------------------|
| 037 | USA/Canada - CECP |
| 256 | International - #1 |
| 273 | Germany F.R./Austria - CECP |
| 274 | Belgium - CECP |
| 275 | Brazil - CECP |
| 276 | Canada (French) - 94 |
| 277 | Denmark, Norway - CECP |
| 278 | Finland, Sweden - CECP |
| 280 | Italy - CECP |
| 281 | Japan (Latin) - CECP |
| 282 | Portugal - CECP |
| 283 | Spain - 190 |
| 284 | Spain/Latin America - CECP |
| 285 | United Kingdom - CECP |
| 297 | France - CECP |
| 423 | Greece |
| 424 | Israel (Hebrew) |
| 500 | International - #5 |
| 871 | Iceland - CECP |

If a document uses an unsupported code page (for example, 905 Turkey or 880 Cyrillic), use the print options to print the document correctly.

Work Station Support

When a document is created, it assumes the code page of the document or text profile from which it was created. To set a text profile to a desired code page, it must be created on a work station that is configured to the system as that code page (CHRID parameter on CRTDEV DSP command). Once a document is created, it continues to exist in that code page regardless of the work station on which it is viewed or edited, and regardless of the code page of that work station.

When a document is edited or viewed on a work station configured as a different code page than the document, and both the document and work station code pages are in the list of supported word processing code pages, the text is translated to the code page of the work station so it displays correctly. When the text is edited, the text is translated back to the code page of the document so that it continues to be stored in the correct code page. If either the document or work station code page is not one of the supported code pages, no translation takes place and the document may display incorrectly. Thus, documents created in an unsupported code page should only be edited or viewed on a work station that is also configured as that unsupported code page.

Document Interchange

Normally when a document is sent to another system, the character set and code page used are 337 256. This allows for the best interchange between the AS/400 system and other IBM office products. Thus, when the document exists in one of the code pages supported by word processing, it is translated to 337 256 when it is sent to another system. If the document exists in an unsupported code page (for example, 905 Turkey), it is sent in that code page, and only displays properly if displayed on a work station configured with that code page.

When a document is received on the AS/400 system, and both the document code page and the code page specified in the system value QCHRID are supported code pages, it is translated to the code page specified in the system value QCHRID for that system. This is done because it is assumed that the majority of work stations on that system are in that code page and no translation needs to occur for those work stations when the document is viewed or edited. Once the document is on the system, it is treated the same as a document created on the AS/400 system.

Appendix B. How to Create Control Language Programs and Commands

This appendix summarizes how to create, compile, run, and debug control language (CL) programs and create commands. It is intended as a starting point for a beginning programmer or as a reference for the experienced programmer. Refer to the *CL Programmer's Guide* for more detailed information on creating CL programs.

Creating Control Language Programs

This section describes how to create CL programs.

Creating a Source Program

Use the SEU editor to create the source code. To do this, use either the STRSEU or the STRPDM command.

The usual source file used for CL commands is QCLSRC in your own library. If you do not have this file, create it with the following command:

```
CRTSRCPF FILE(library-name/QCLSRC)
```

When you create the CL source program (with either STRSEU or STRPDM), specify CLP for the member type. For example, to create the program PROGRAM1 in file QCLSRC of library OFCLIB using SEU, you would use the command:

```
STRSEU SRCFILE(OFCLIB/QCLSRC) SRCMBR(PROGRAM1) TYPE(CLP)
```

This command creates the new member PROGRAM1, and takes you to the text editor to allow you to edit the new member.

Editing

If you do not know the parameter for a particular command while editing, you can type the command that you want to put in the program and press F4. Fill in the parameter values and this is entered into your program.

If you want uppercase and lowercase characters for documentation in your program while editing, press F13 and take option N to uppercase input only parameter.

While editing, press the Help key to get help on the functions provided by SEU.

To exit editing, press F3.

Some CL Program Statements

You are required to have the PGM statement as the first command and the ENDPGM statement as the last command of all CL programs.

The /* */ is for comments and can be on a line by itself or at the end of a statement.

The RETURN command returns control to the calling program.

The GOTO statement is used for loops or to go anywhere else in the program. Labels are used for the GOTO statement. Labels can be up to 10 characters followed by a colon (:).

When using the IF statement, be careful when using comments. Insert the comment before the IF statement to avoid any confusion with the IF logic. Use the DO and DOEND statements around logic that has more than one statement within the IF statement.

File Considerations

You can declare a file in your program with the DCLF statement. This can be a system file or a user file. When the program is compiled, the fields are created and can be used by the program. Other fields not declared by the file must be specifically declared by the DCL statement.

The file in the DCLF statement must exist before compiling the program. If you are creating a file in your program (for example, QRYDST to an output file), you can declare the system file that has the same layout the command produces (QRYDST uses QSYS/QAOSILIN for incoming distributions). After you create your own file, you can use the OVRDBF statement to use your newly created file (it has the same layout as the system file). In this way you can compile your program without the file that is actually used existing.

Files are automatically opened and closed. However, if you use more than one file in your program, you should open and close them yourself.

Records in a CL file are given sequentially. Use the RCVF statement to do this. The first RCVF gets the first record, the next RCVF gets the next, and so on. You should monitor for message CPF0864 (End of File) to detect reading past the end of the file when in a loop.

Compiling the Program

After editing your program, you need to compile it. The compiled program can reside in a different library than the source. You can use the CRTCLPGM command to compile the program, or option 14 in PDM. With option 14 in PDM, if you press PF4 after typing 14, you can get the compile option up.

For example, to compile the source program PROGRAM1 in source file QCLSRC of library OFCLIB and you want the compiled program in library OFCLIB2, use the CRTCLPGM command:

```
CRTCLPGM PGM(OFCLIB2/PROGRAM1) SRCFILE(OFCLIB/QCLSRC)
```

Printed Output of Compiled Program

After you compile your program, it is written to the output queue. You can view your compile with DSPJOB option 4 if you compiled with CRTCLPGM. If you compiled through PDM, you get a batch job number. Use that number in the DSPJOB command to see your output or the status for that job.

If you have an unsuccessful compilation, the errors are listed at the bottom. If the compilation is successful, you get an object with the program name, specified of type *PGM, in the library specified.

Running Your Program

To run your program, use the CALL command. If you have parameters to send to the program, use the PARM parameter to specify them.

A more user-friendly way to run your program is from a command. Creating commands is discussed later in this section. You can also use a command in a menu. Use STRSDA to create a user menu.

Debugging Your Program

Sometimes you have bugs in your program. To find them conveniently, use the debugging tool provided on the system. You can stop on any statement in your program and display a program variable there.

Common commands used for debugging are:

- STRDBG** Use the STRDBG command and specify the program you are debugging. Specify UPDPROD(*YES) if creating files in your program.
- ADDBKP** Specify the statement number (get your statement numbers from the output spooled file in your output queue). You can also specify the variables you want to see (you need to prefix them with the ampersand (&) symbol for CL variables).
- DSPPGMVAR** After your program reaches a stop, you can press F10 to get to the command line. From there you can use DSPPGMVAR to display a program variable (remember to use the ampersand symbol (&) before the variable name). You can also type a command on this command line.
- ENDDBG** After you are finished debugging, use this command to end the debug mode.

See the *CL Programmer's Guide* for more information on debugging.

Creating Physical Files

Sometimes you need a file that is user created and not system created. To create the physical file, you need a source file with data description specifications (DDS). The convention is to put DDS source in the file with the name QDDSSRC. The type is PF.

See the *DDS Reference* manual for more information about DDS.

Example DDS Source File

Following is an example of a DDS source file:

```
*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ..
      R PERSR
A      EMP#           8 0
A      FNAME          15
A      LNAME          20
A      ADDR1          20
A      CITY           12
A      STATE          2
A      ZIP             9 0
A      DEPT            8
A      PHONE          15
A      K EMP#
```

To create the physical file in your library, use the CRTPF command and specify the source file where your DDS exists. Then compile your CL program where you use your own file.

To get records out to the file, you can write a high-level language program (CL does not support writing data to a file), or you can use the STRDFU command.

Example CL Program

The following is an example CL program based on Figure 4-9 on page 4-38. You can find out more about each command in this program by referring to the *CL Reference* manual.

```
PGM          PARM(&USRID &ADDR)
/* This statement is the PGM statement and there are */
/* two parameters that need to be sent to it and */
/* they are &USRID and &ADDR. These are declared */
/* by the DCL statement. */

/* The DCL statement tells what the variable names are*/
/* what type they are and the length.
DCL          VAR(&USRID) TYPE(*CHAR) LEN(8)
DCL          VAR(&ADDR) TYPE(*CHAR) LEN(8)
DCL          VAR(&FILENAM) TYPE(*CHAR) LEN(8)
DCL          VAR(&DOCNAME) TYPE(*CHAR) LEN(7)

/* The declare file statement declares a file that */
/* already exists on the system. In this case it is */
/* the system file supplied with the system for the */
/* output file that QRYDST outputs for incoming */
/* distributions. See the Database Guide for */
/* information about office output files. */
DCLF        FILE(QSYS/QAOSILIN) RCDfmt(OSLIN) /* System +
          QRYDST outfile layout */
```

Figure B-1 (Part 1 of 3). A Program to Receive Mail

```

/* The '+' plus symbol at the end of the previous */
/* statement indicates continuation of the command. */

/* This command creates an outfile in library QTEMP. */
QRYDST      USRID(&USRID &ADDR) OUTFILE(QTEMP/QRYDSTIN)

/* OVRDBF will override the system file QAOSILIN with */
/* the file QRYDSTIN that was just created by QRYDST. */
OVRDBF      FILE(QAOSILIN) TOFILE(QTEMP/QRYDSTIN)

/* Each of the records in file QRYDSTIN will be read */
/* sequentially by the RCVF because it is in a loop. */
/* The label that is the beginning of the loop is */
/* READF. The GOTO statement later defines where the */
/* loop ends. */
READF:      RCVF          RCD_FMT(OSLIN)

/* The next statement monitors for a message from the */
/* RCVF command. If the message CPF0864 (End of File)*/
/* is issued, the program calls the 'GOTO' */
/* statement and goes to the label EOF at the end of */
/* the program. */
MONMSG      MSGID(CPF0864) EXEC(GOTO CMDLBL(EOF))

/* The next statement is an IF statement. The DO and */
/* ENDDO statement are used to show the logic more */
/* clearly. */
/* The &LINDTP is a variable that is declared in the */
/* system file QAOSILIN. It is expanded when the */
/* program is compiled. */
IF COND(&LINDTP *EQ 0) THEN( +
DO)
    RCV DSTID(&LINDID) +
        USRID(&LINRUI &LINRUA) DOC(*NONE) +
        OUTFILE(&USRID/MSGFILE) +
        OUTMBR(*FIRST *ADD) OUTDTATYP(*ALL)
ENDDO
ELSE +
DO
/* You can use a comment statement here. Just do */
/* not use it after the IF statement. */
IF COND( (&LINDTP *EQ 2) *OR (&LINDTP *EQ 11) ) +
THEN( DO )
    /* The next statement changes the variable */
    /* &DOCNAME to the value 'DOC' concatenated */
    /* with the characters in position 17, 18, */
    /* 19 and 20 in the &LINDID field. In */
    /* other words, &DOCNAME is changed to 'DOC' */
    /* concatenated with the substring of field */
    /* &LINDID field of position 17 for 4 */
    /* characters. */

```

Figure B-1 (Part 2 of 3). A Program to Receive Mail

```

        CHGVAR VAR(&DOCNAME) VALUE( +
            'DOC' *CAT %SST(&LINDID 17 4) )
        RCV DSTID(&LINDID) +
            USRID(&LINRUI &LINRUA) +
            DOC(&DOCNAME) +
            FLR(&USRID)
    ENDDO
    ELSE +
    DO
        CHGVAR VAR(&FILENAM) VALUE( +
            'FILE' *CAT %SST(&LINDID 17 4) )
        RCV DSTID(&LINDID) USRID(&LINRUI &LINRUA) +
            OUTFILE(&USRID/&FILENAM) +
            OUTMBR(*FIRST *ADD) OUTDTATYP(*ALL)
    ENDDO
    /* The following ENDDO statement ends the DO          */
    /* after the ELSE statement.                          */
    ENDDO

    /* The GOTO statement goes to the label READF to     */
    /* complete the loop. This will read all of the     */
    /* records in the QRYDST outfile sequentially until */
    /* end of file is reached.                          */
    GOTO      CMDLBL(READF)

EOF:    /* This statement is where control goes to when the */
        /* end of file message is monitored for by the    */
        /* MONMSG command. The label is EOF. When control */
        /* gets to here the temporary QRYDST outfile is  */
        /* deleted (QRYDSTIN in library QTEMP).          */
        DLTFFILE(QTEMP/QRYDSTIN)

        /* The program then returns to the calling program */
        /* with the RETURN statement.                      */
        RETURN

        /* The ENDPGM is the last statement and is necessary */
        /* for a CL program.                                */
        ENDPGM

```

Figure B-1 (Part 3 of 3). A Program to Receive Mail

Creating Control Language Commands

Commands are an interface for calling programs that have parameters. You can get help by pressing F4. The parameters are displayed.

Creating Source

First, create the command source with an editor. The conventional source physical file to use is QCMDSRC in your own library. If this source file does not exist, use the CRTSRCPF command to create it. The type of member to create for a command is CMD. The statements to use in command source are described in the *CL Reference* manual. The most common statements are CMD and PARM.

Example Command Source

To define parameters for the previous example program, the user ID and the address are both 8 characters in length (LEN(8)). They are both required (MIN(1) MAX(1)). If they were optional the values would be MIN(0) MAX(1). Both the user ID and address are character type (TYPE(*CHAR)). The prompt is in the description of each parameter. For an example of the command source, see Figure B-2.

```

/*****
/*
/*  COMMAND NAME: RCVMAIL
/*
/*  COMMAND TITLE: Receive Mail
/*
/*  DESCRIPTION: This command receives messages and documents
/*                for a specified user.
/*
/*
/*****
RCVMAIL: CMD  PROMPT('Receive Mail')
           PARM  KWD(USERID)  +
              TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) MAX(1) +
              VARY(*NO)  EXPR(*YES) PASSATR(*NO) FULL(*NO) +
              PROMPT('User ID')
           PARM  KWD(ADDRESS) +
              TYPE(*CHAR) LEN(8) RSTD(*NO) MIN(1) MAX(1) +
              VARY(*NO)  EXPR(*YES) PASSATR(*NO) FULL(*NO) +
              PROMPT('Address')
```

Figure B-2. Command Source Example

Creating the Command

Use CRTCMD to create the command. Specify the source file to indicate where the command source is (QCMDSRC).

To run the command, do the following:

1. Type the command name on the command line.
2. Do one of the following:
 - a. Press F4 to get help for the parameter values.
 - b. Type the parameters and their values after the command name.

Appendix C. Office Services Output File Considerations

This appendix discusses output files for the commands.

Add Calendar Item (ADDCALITM) Output File Considerations

Use the Add Calendar Item (ADDCALITM) command to optionally produce a database file that contains limited information about each item that you added when using the command. The calendar name (AICAL), calendar owner (AIODEN and AIODGN), and item identifier (AIIID) fields uniquely identify the item for use with other commands (such as DSPCALITM).

The model output file supplied with the system is named QAOYADDITM and is located in library QOFC. Use the following data description specifications to create the model output file:

```
A*** DDS specifications for QOFC/QAOYADDITM *****
A*
A          R ADDITM                TEXT('Record for adding +
A          calendar items (ADDCALITM)')
A          AICEN                    1A    TEXT('Century')
A          AIDAT                     6A    TEXT('Date (YYMMDD)')
A          AITIM                     6A    TEXT('Time (HHMMSS)')
A          AICALC                   4S 0   TEXT('Calendar name character +
A          set')
A          AICALP                   4S 0   TEXT('Calendar name code page')
A          AICAL                    10A   TEXT('Calendar name')
A          AIODEN                    8A    TEXT('Calendar owner user ID')
A          AIODGN                    8A    TEXT('Calendar owner address')
A          AIIID                    20A   TEXT('Item identifier')
A          AIDATE                    6A    TEXT('Item date (YYMMDD)')
A          AISTIM                    5A    TEXT('Item start time (HH:MM)')
A          AIETIM                    5A    TEXT('Item end time (HH:MM)')
A*** End of Specifications *****
```

Note: On double-byte systems the field AICAL has a data type of O (open). This field can contain mixed single-byte and double-byte data on these systems.

The meaning of each field in the record is:

- *AICEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *AIDAT* (Job date). The date the output file was created. Specify the date as YYMMDD, where YY=year, MM=month, and DD=day.
- *AITIM* (Job time). The time the output file was created. Specify the time as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *AICALC* (Calendar name character set). The calendar name field (AICAL), which is stored in the output file in this character set.
- *AICALP* (Calendar name code page). The calendar name field (AICAL), which is stored in the output file in this code page.
- *AICAL* (Calendar name). The 1- to 10-character name assigned to the calendar when the user created it.

- *AIDEN* (Calendar owner user ID). The user ID of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *AIDGN* (Calendar owner address). The address of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *AIID* (Item identifier). The item identifier, is a 20-character string which, when combined with the calendar name and owner, can uniquely identify an item. Use the item identifier as input to other commands (such as *DSPCALITM*) to process this specific item.
- *AIDATE* (Item date). The date the indicated on the calendar when the calendar item is scheduled. Specify the date *YYMMDD*, where *YY*=year, *MM*=month, and *DD*=day.
- *AISTIM* (Item start time). The time indicated on the calendar when the calendar item is scheduled to start. Specify the time as *HH:MM*, where *HH*=hour and *MM*=minutes. This field is blank if the item is a reminder, because reminders do not have a start time.
- *AJETIM* (Item end time). The time indicated on the calendar when the calendar item is scheduled to end. Specify the time as *HH:MM*, where *HH*=hour and *MM*=minutes. This field is blank if the item is a reminder, job, or System/36 procedure, because these items do not have an end time.

Display Calendar Authority (DSPCALAUT) Output File Considerations

Use the Display Calendar Authority (DSPCALAUT) command to create a database file that contains information about calendar authorities. Each calendar contains one public authority record and possibly one or more specific authority records.

When the command is used by someone other than the calendar owner or a user with **SECADM* or **ALLOBJ* special authority, only the public authority record and the current user's specific authority record are written to the output file. Otherwise, the public authority record and all specific authority records are written.

The authority fields (CAxxxA) can contain the values **CHANGE*, **ADDITEM*, **USE*, **TIMES*, or **EXCLUDE*.

The model output file supplied with the system is named QAOYCALAUT and is located in library QOFC. Use the following data description specifications to create the model output file:

```

A***  DDS specifications for QOFC/QAOYCALAUT  *****
A*
A      R CALAUT                                TEXT('Record for displaying +
A                                         calendar authorities (DSPCALAUT)')
A      CACEN 1A                               TEXT('Century')
A      CADAT 6A                               TEXT('Date (YYMMDD)')
A      CATIM 6A                               TEXT('Time (HHMMSS)')
A      CACALC 4S 0                            TEXT('Calendar name character +
A                                         set')
A      CACALP 4S 0                            TEXT('Calendar name code page')
A      CACAL 10A                              TEXT('Calendar name')
A      CAODEN 8A                              TEXT('Calendar owner +
A                                         (user)')
A      CAODGN 8A                              TEXT('Calendar owner +
A                                         (address)')
A      CAUDEN 8A                              TEXT('User ID +
A                                         (or *PUBLIC)')
A      CAUDGN 8A                              TEXT('User address')
A      CAUNCA 8A                              TEXT('Unclassified +
A                                         authority')
A      CACONA 8A                              TEXT('Confidential +
A                                         authority')
A      CAPRSA 8A                              TEXT('Personal authority')
A***  End of Specifications  *****

```

Note: On double-byte systems, the field CACAL has a data type of O (open). This field can contain mixed single-byte and double-byte data on these systems.

The meaning of each field in the record is:

- *CACEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *CADAT* (Job date). The date the output file was created. Specify the date as YYMMDD, where YY=year, MM=month, and DD=day.
- *CATIM* (Job time). The time the output file was created. Specify the time as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *CACALC* (Calendar name character set). The calendar name field (*CACAL*), which is stored in the output file in this character set.
- *CACALP* (Calendar name code page). The calendar name field (*CACAL*), which is stored in the output file in this code page.
- *CACAL* (Calendar name). The 1- to 10-character name assigned to the calendar when the user created it.
- *CAODEN* (Calendar owner user ID). The user ID of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *CAODGN* (Calendar owner address). The address of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *CAUDEN* (Authorized user's user ID). The user ID of the user authorized to the calendar. This is stored in the output file in character set 930, code page 500. This field can also contain the special value *PUBLIC, meaning the

authority record is for the public authority given to the calendar and not for a specific user.

- *CAUDGN* (Authorized user's address). The address of the user authorized to the calendar. This is stored in the output file in character set 930, code page 500. This field is blank if the user ID field (*CAUDEN*) contains the special value **PUBLIC*.
- *CAUNCA* (Unclassified authority). The authority given to the user specified in the authorized user's fields (*CAUDEN* and *CAUDGN*) for unclassified items on the calendar.
- *CACONA* (Confidential authority). The authority given to the user specified in the authorized user's fields (*CAUDEN* and *CAUDGN*) for confidential items on the calendar.
- *CAPRSA* (Personal authority). The authority given to the user specified in the authorized user's fields (*CAUDEN* and *CAUDGN*) for personal items on the calendar.

Display Calendar Description (DSPCALD) Output File Considerations

Use the Display Calendar Description (DSPCALD) command to create a database file that contains information about calendars (such as name, owner, and working times).

The authority fields (*DCxxAU*) can contain the values **CHANGE*, **ADDITEM*, **USE*, **TIMES*, or **EXCLUDE*.

The following fields are blank if the record is added by a user who has only **EXCLUDE* authority to the calendar: *DCMDEN*, *DCMDGN*, *DCNMGR*, *DCNMOD*, *DCRTIM*, *DCARMD*, *DCRMOD*, *DCWTIM*, and *DCAUTO*.

The following fields are blank if the record is added by a user who has only **TIMES* authority to the calendar: *DCMDEN*, *DCMDGN*, *DCNMGR*, *DCNMOD*, *DCRTIM*, *DCARMD*, *DCRMOD*, and *DCAUTO*.

The model output file supplied with the system is named QAOYDSPCAL and is located in library QOFC. Use the following data description specifications to create the model output file:

```

A***   DDS specifications for QOFC/QAOYDSPCAL   *****
A*                                           *
A           R DSPCAL                       TEXT('Record for displaying +
A                                           calendar descriptions +
A                                           (DSPCALD)')
A           DCCEN           1A             TEXT('Century')
A           DCDAT           6A             TEXT('Date (YYMMDD)')
A           DCTIM           6A             TEXT('Time (HHMMSS)')
A           DCCALC          4S 0           TEXT('Calendar name character +
A                                           set')
A           DCCALP          4S 0           TEXT('Calendar name code page')
A           DCCAL           10A            TEXT('Calendar name')
A           DCODEN          8A             TEXT('Calendar owner user ID')
A           DCODGN          8A             TEXT('Calendar owner address')
A           DCSCHD          1A             TEXT('Scheduling Calendar +
A                                           flag')
A           DCMDEN          8A             TEXT('Calendar manager user +
A                                           ID')
A           DCMDGN          8A             TEXT('Calendar manager address')
A           DCNMGR          4A             TEXT('Notify manager flag')
A           DCNMOD          5A             TEXT('Notify manager mode')
A           DCRTIM          3S 0           TEXT('Reminder lead time')
A           DCARMOD         4A             TEXT('Automatic reminder flag')
A           DCRMOD          5A             TEXT('Reminder mode')
A           DCTXT           50A            TEXT('Calendar description')
A           DCTCTC          4S 0           TEXT('Calendar description +
A                                           character set')
A           DCTXTP          4S 0           TEXT('Calendar description +
A                                           code page')
A           DCWTIM          672A           TEXT('Calendar working +
A                                           times')
A           DCPUUAU         8A             TEXT('Public unclassified +
A                                           authority')
A           DCPCAU          8A             TEXT('Public confidential +
A                                           authority')
A           DCPPAU          8A             TEXT('Public personal +
A                                           authority')
A           DCUDEN          8A             TEXT('Current user ID')
A           DCUDGN          8A             TEXT('Address')
A           DCUUUAU         8A             TEXT('User unclassified +
A                                           authority')
A           DCUCAU          8A             TEXT('User confidential +
A                                           authority')
A           DCUPAU          8A             TEXT('User personal +
A                                           authority')
A           DCAUTO          10A            TEXT('Automatically +
A                                           process notices')
A
A***   End of Specifications   *****

```

Note: On double-byte systems the fields DCCAL and DCTXT have a data type of O (open). These fields can contain mixed single-byte and double-byte data on these systems.

The meaning of each field in the record is:

- *DCCEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *DCDAT* (Job date). The date the output file was created. Specify the date as YYMMDD, where YY=year, MM=month, and DD=day.
- *DCTIM* (Job time). The time the output file was created. Specify the time as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *DCCALC* (Calendar name character set). The calendar name field (DCCAL), which is stored in the output file in this character set.
- *DCCALP* (Calendar name code page). The calendar name field (DCCAL), which is stored in the output file in this code page.
- *DCCAL* (Calendar name). The 1- to 10-character name assigned to the calendar when the user created it.
- *DCODEN* (Calendar owner user ID). The user ID of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *DCODGN* (Calendar owner address). The address of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *DCSCHD* (Scheduling calendar flag). The flag that indicates whether the calendar in the calendar name field (DCCAL) is the scheduling calendar for the owner specified in the owner fields (DCODEN and DCODGN). The flag is set to Y if it is the owner's scheduling calendar; otherwise it is set to N.
- *DCMDEN* (Calendar manager user ID). The user ID of the calendar's manager. This is stored in the output file in character set 930, code page 500. The manager is the user who manages the items on the calendar.
- *DCMDGN* (Calendar manager address). The address of the calendar's manager. This is stored in the output file in character set 930, code page 500. The manager is the user who manages the items on the calendar.
- *DCNMGR* (Notify manager flag). The flag that indicates whether the calendar manager in the manager fields (DCMDEN and DCMDGN) is notified of changes made to the calendar. The flag is set to *YES if the manager is to be notified; otherwise it is set to *NO.
- *DCNMOD* (Notify manager mode). The method for notifying the calendar manager named in the manager fields (DCMDEN and DCMDGN) of changes made to the calendar. The value is either *MSGQ (notification is received only on the manager's message queue) or *MAIL (notification is received on both the manager's message queue and in the manager's mail). This field is blank if DCNMGR is set to *NO.
- *DCRTIM* (Reminder lead time). The default number of minutes that a reminder message should be sent.
- *DCARM* (Automatic reminder flag). The flag that indicates the default for whether a reminder message should be sent for items added to the calendar. The flag is set to *YES if reminder messages should be sent by default and *NO if reminder messages should not be sent by default.
- *DCRM* (Reminder message mode). The method for sending reminder messages for items on the calendar. The value is either *MSGQ (reminder messages sent only to calendar owner's message queue) or *MAIL (reminder

messages sent to the calendar owner's message queue and to the owner's mail). This field is blank if DCARMD is set to *NO.

- *DCTXT* (Calendar description). The description associated with the calendar.
- *DCTCTC* (Calendar description character set). The calendar description field (*DCTXT*), which is stored in the output file in this character set.
- *DCTXTP* (Calendar description code page). The calendar description field (*DCTXT*), which is stored in the output file in this code page.
- *DCWTIM* (Calendar working times). The user-defined working times for the calendar. Each byte represents 15 minutes, starting on Monday at 12 a.m. and ending on Sunday at 11:45 p.m. Each byte contains an X if the calendar has that 15 minutes defined as a working time; otherwise the byte is blank.
- *DCPUAU* (Public unclassified authority). The authority given to *PUBLIC for unclassified items on the calendar.
- *DCPCAU* (Public confidential authority). The authority given to *PUBLIC for confidential items on the calendar.
- *DCPPAU* (Public personal authority). The authority given to *PUBLIC for personal items on the calendar.
- *DCUDEN* (Current user's user ID). The user ID of the user who added this record to the output file. This is stored in the output file in character set 930, code page 500. This field is blank if the user who added this record is not in the system distribution directory.
- *DCUDGN* (Current user's address). The address of the user who added this record to the output file. This is stored in the output file in character set 930, code page 500. This field is blank if the user who added this record is not in the system distribution directory.
- *DCUUAU* (User's unclassified authority). The authority given to the user in the current user's fields (*DCUDEN* and *DCUDGN*) for unclassified items on the calendar.
- *DCUCAU* (User's confidential authority). The authority given to the user in the current user's fields (*DCUDEN* and *DCUDGN*) for confidential items on the calendar.
- *DCUPAU* (User's personal authority). The authority given to the user in the current user's fields (*DCUDEN* and *DCUDGN*) for personal items on the calendar.
- *DCAUTO* (Automatically process notices). The times when meeting notices for this calendar are processed automatically. The possible values are *NONE (notices are never processed automatically), *SPECIFIC (notices are processed automatically only if the meeting scheduler is authorized to the calendar), and *ALL (notices are always processed automatically).

Display Calendar Item (DSPCALITM) Output File Considerations

Use the Display Calendar Item (DSPCALITM) command to produce a database file that contains complete information about calendar items.

When used by a user with only *TIMES authority to the item, the only record type added to the output file is the base record (DIRTYP = 0001). The remaining data

field (DIRDTA) of this record contains only the item type, item date, item start time, and item end time. The remaining portion of the field is blank.

The model output file supplied with the system is named QAOYDSPITM and is located in library QOFC. Use the following data description specifications to create the model output file:

```
A*** DDS specifications for QOFC/QAOYDSPITM *****
A*
A          R DSPITM                TEXT('Record for displaying +
A          calendar items (DSPCALITM)')
A          DICEN                    1A    TEXT('Century')
A          DIDAT                    6A    TEXT('Date (YYMMDD)')
A          DITIM                    6A    TEXT('Time (HHMMSS)')
A          DICALC                   4S 0  TEXT('Calendar name character +
A          set')
A          DICALP                   4S 0  TEXT('Calendar name code page')
A          DICAL                    10A   TEXT('Calendar name')
A          DIODEN                   8A    TEXT('Calendar owner user ID')
A          DIODGN                   8A    TEXT('Calendar owner address')
A          DIIID                    20A   TEXT('Item identifier')
A          DIRTYP                   4A    TEXT('Record type')
A          DIRDTA                   277A  TEXT('Remaining record data')
A*** End of specifications *****
```

Note: On double-byte systems the fields DICAL and DIRDTA has a data type of O (open). These fields can contain mixed single-byte and double-byte data on these systems.

The meaning of each field in the record is:

- *DICEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *DIDAT* (Job date). The date the output file was created. Specify the date as YYMMDD, where YY=year, MM=month, and DD=day.
- *DITIM* (Job time). The time the output file was created. Specify the time as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *DICALC* (Calendar name character set). The calendar name field (DICAL), which is stored in the output file in this character set.
- *DICALP* (Calendar name code page). The calendar name field (DICAL), which is stored in the output file in this code page.
- *DICAL* (Calendar name). The 1- to 10-character name assigned to the calendar when the user created it.
- *DIODEN* (Calendar owner user ID). The user ID of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *DIODGN* (Calendar owner address). The address of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *DIIID* (Item identifier). The item identifier, a 20-character string which when combined with the calendar name and owner, can uniquely identify an item.
- *DIRTYP* (Record type). The record type that identifies the format the remaining data field (DIRDTA) is in. Possible values are: 0001=base record, 0002=text record, 0003=invitee record, and 0004=job or procedure record.
- *DIRDTA* (Remaining record data). The remaining data, which varies in format based on the record type field (DIRTYP). See the following table for various formats.

| Record Code | Data Field | Description |
|-------------|------------|---|
| 0001 | DIRDTA | <ul style="list-style-type: none"> • Item type (position 1-9, alphanumeric). The type of calendar item. Possible values are *REMINDER, *EVENT, *MEETNG, *JOB, or *S36PRC. • Item date (position 10-15, alphanumeric). The date the calendar item is scheduled on the calendar. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day. • Item start time (position 16-20, alphanumeric). The time on the calendar when the calendar item is scheduled to start. The time is specified as HH:MM, where HH=hour and MM=minutes. This field is blank if the item is a reminder, because reminders do not have a start time. • Item end time (position 21-25, alphanumeric). The time indicated on the calendar when the calendar item is scheduled to end. The time is specified as HH:MM, where HH=hour, and MM=minutes. This field is blank if the item is a reminder, job, or System/36 procedure, because these items do not have an end time. • Item security (position 26-38, alphanumeric). The item security level. Possible values are *UNCLASSIFIED, *CONFIDENTIAL, or *PERSONAL. If the item is a meeting, then this security value is for the meeting entry. • Item meeting security (position 39-51, alphanumeric). The meeting item security level. Possible values are *UNCLASSIFIED, *CONFIDENTIAL, or *PERSONAL. If the item is not a meeting or the meeting has been deleted (item status = *DELETED), this field is blank. • Reminder message (position 52-55, alphanumeric). If the item is an event, reminder, or meeting, this flag indicates whether a reminder message is sent for the item. The flag is set to *YES if a reminder message is sent and *NO if a reminder message is not sent (or has already been sent). For jobs and System/36 procedures, this field is blank. • Reminder message date (position 56-61, alphanumeric). The date the reminder message for this item is scheduled to be sent. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day. This field is blank when the item is a job or System/36 procedure, or when the reminder message flag is set to *NO. • Reminder message time (position 62-66, alphanumeric). The time the reminder message for this item is scheduled to be sent. The time is specified as HH:MM, where HH=hour and MM=minutes. This field is blank when the item is a job or System/36 procedure, or when the reminder message flag is set to *NO. • Item status or job pending flag (position 67-76, alphanumeric). Meeting or event item status. If the item is an event, this field is either *TENTATIVE or *CONFIRMED. If the item is a meeting, this field is *TENTATIVE, *CONFIRMED, *POSTPONED, *CANCELED, or *DELETED. If the item is a job or System/36 procedure, this flag indicates whether the job or procedure is still pending (if it has not been submitted). The flag is set to *YES if the job or procedure is has not yet been submitted and *NO if the job or procedure has already been submitted. For reminders, this field is blank. • Record number for invitee notes (position 77-80, zoned(4,0)). Contains the record number of the first text record that is part of the meeting entry notes. All text records before this record are part of the meeting text (subject, place, and purpose). This field is zero for non-meeting items and for meetings without any meeting entry notes. |
| 0002 | DIRDTA | <ul style="list-style-type: none"> • Text record number (position 1-4, zoned(4,0)). The record number for this text line. • Text character set (position 5-8, zoned(4,0)). The text line stored in the output file in this character set. • Text code page (position 9-12, zoned(4,0)). The text line stored in the output file in this code page. • Text line (position 13-62, alphanumeric or open). The line of text for the item. |

| Record Code | Data Field | Description |
|-------------|------------|--|
| 0003 | DIRDTA | <ul style="list-style-type: none"> • Invitee calendar character set (position 1-4, zoned(4,0)). The invitee calendar name is stored in the output file in this character set. • Invitee calendar code page (position 5-8, zoned(4,0)). The invitee calendar name is stored in the output file in this code page. • Invitee calendar name (position 9-18, alphanumeric or open). The calendar name invited to the meeting (may be blank if the invitee is on a remote system and was invited by user ID and address only). • Invitee calendar owner user ID (position 19-26, alphanumeric). The user ID of the invitee calendar's owner. This is stored in the output file in code page 500, character set 930. • Invitee calendar owner address (position 27-34, alphanumeric). The address of the invitee calendar's owner. This is always stored in the output file in code page 500, character set 930. • Invitee status (position 35-47, alphanumeric). The invitee's meeting status. Possible values are *UNKNOWN, *ATTENDING, *NOTATTENDING, *ALTERNATE, or *NOTINVITED. |
| 0004 | DIRDTA | <ul style="list-style-type: none"> • Job description or System/36 procedure name (position 1-10, alphanumeric). If the item is a job, this field contains the name of the job description to be used when the job is submitted. If the item is a System/36 procedure, this field contains the name of the procedure to be submitted. • Job description library or System/36 procedure library (position 11-20, alphanumeric). If the item is a job, this field contains the library where the job description is located. If the item is a System/36 procedure, this field contains the library where the procedure is located. • System/36 procedure priority (position 21-21, alphanumeric). If the item is a System/36 procedure, this field contains the priority level at which to submit the procedure. For jobs, this field is blank. • Job command or System/36 procedure parameters (position 22-277, alphanumeric). If the item is a job, this field contains the job command string to be submitted. If the item is a System/36 procedure, this field contains the parameter string to submit with the procedure. |

Display Directory (DSPDIR) Output File Considerations

Use the Display Directory (DSPDIR) command to create a database file that contains the information from the system distribution directory. Use the OUTDTA parameter to specify the type of data to place in an output file (local only or local and shadowed only). You can specify three types of output files with the OUTFILFMT parameter. The default is OUTFILFMT(*TYPE1), which uses the format modeled in file QAOSDIRO in library QSYS. The second type is specified by OUTFILFMT(*TYPE2), and the third is specified by OUTFILFMT(*TYPE3).

When OUTFILFMT(*TYPE1) is specified, the output file contains either basic or full information about the directory entries depending on the USRID, USER, and DETAIL parameters.

If the USRID or USER parameters specified a unique entry, one detail database record is written with the full directory details. If more than one description exists for the specified USRID or USER, the output file will contain a full detail record for each description.

When the USRID specifies, or takes the default value of *ALL, the DETAIL parameter will determine what data will be written to the database file. If DETAIL(*BASIC) was specified, the database file will only contain the user ID, address, and description. If DETAIL(*FULL) was specified, a full detail record is built for each description. Note that when multiple descriptions exist in the directory for a given user ID and address, one record will be written for each description. The only data that is different between these records is the description field.

When OUTFILFMT(*TYPE2) is specified, the data is sent to the output file in a different format because of field length changes, and additional fields.

If DETAIL(*BASIC) is specified, then the model output file used is QAOSDIRB in library QSYS. The records contain the user ID, address, and description. If DETAIL(*FULL) or a specific user is specified, the model output file used is QAOSDIRF in library QSYS. A full detail record is built with all the additional fields included.

When OUTFILFMT(*TYPE3) is specified, the output file contains either BASIC or FULL information about file formats.

If DETAIL(*BASIC) is specified, an output file of model format QAOSDIRB in library QSYS is generated. This file has the basic values. If DETAIL(*FULL) is specified, an output file of model format QAOSDIRX in library QSYS is generated. This file has all the directory fields and the X.400 files.

The following data description specifications were used to create the model output file when OUTFILFMT (*TYPE1) is specified:

```

A* Start of specifications
A*
A* File Name: QAOSDIRO in library QSYS
A* File Type: Physical
A* Record Format Name: OSDIRE
A*
A*
A          R OSDIRE
A          WOSDDEN      8A          TEXT('User ID')
A          WOSDDGN      8A          TEXT('Address')
A          WOSDDESC     50A         TEXT('User description')
A          WOSDREN      8A          TEXT('System name')
A          WOSDRGN      8A          TEXT('System group')
A          WOSDUSRP     10A         TEXT('User profile')
A          WOSDINDU      1A         TEXT('Indirect user flag')
A          WOSDRCPV      1A         TEXT('Print personal mail')
A          WOSDADD1     36A         TEXT('Mailing address 1')
A          WOSDADD2     36A         TEXT('Mailing address 2')
A          WOSDADD3     36A         TEXT('Mailing address 3')
A          WOSDADD4     36A         TEXT('Mailing address 4')
A          WOSDLOC      36A         TEXT('Location')
A          WOSDTEL1     25A         TEXT('Telephone number 1')
A          WOSDTEL2     25A         TEXT('Telephone number 2')
A          WOSDTEXT     50A         TEXT('Misc. text')
A          WOSDCS       8A          TEXT('System the outfile +
                                     was created on')
A          WOSDCC       1A          TEXT('Century the outfile +
                                     was created')
A          WOSDCD       6A          TEXT('Date the outfile +
                                     was created')
A          WOSDCT       6A          TEXT('Time the outfile +
                                     was created')
A          WOSDFAX     32A         TEXT('Fax telephone +
                                     number')
A*
A* End of specifications

```

Figure C-1. Output File OUTFILFMT (*TYPE1) Example

The meaning of each field in the record is:

- **WOSDDEN** (User ID). The user identification for which the detailed information is provided. This field is put into code from code page 500, character set 930. The user ID along with the address uniquely identifies the directory entry.
- **WOSDDGN** (Address). The address for which the detailed information is provided. This field is put into code from code page 500, character set 930. The address along with the user ID uniquely identifies the directory entry.
- **WOSDDESC** (User description). The user description associated with the user ID and address.
- **WOSDREN** (System name). The system name (known to SNADS) of the system on which this user works. This field is put into code from code page 500/character set 930. The system name along with the system group uniquely identifies a system.

- *WOSDRGN* (System group). The system group (known to SNADS) of the system on which this user works. This field is put into code from code page 500/character set 930. The system group can be blank. The system group along with the system name uniquely identifies a system.
- *WOSDUSRP* (User profile). The system profile name for the user.
- *WOSDINDU* (Indirect user flag). A 1-character flag to indicate whether or not the user is an indirect user (one who does not use the system to receive mail, but receives printed mail). The value 1 indicates the user is an indirect user. The value 0 indicates that the user is not an indirect user.
- *WOSDRCPV* (Receive personal mail). A 1-character flag to specify whether an indirect user is to have personal mail printed. The value 1 indicates the indirect user is to receive personal mail. The value 0 indicates that no personal mail will be printed for the indirect user. The data in this field has no meaning if this is not an indirect user.
- *WOSDADD1* (Address line 1). The first address line for the user. This field is optional input to the directory and can contain blanks.
- *WOSDADD2* (Address line 2). The second address line for the user. This field is optional input to the directory and can contain blanks.
- *WOSDADD3* (Address line 3). The third address line for the user. This field is optional input to the directory and can contain blanks.
- *WOSDADD4* (Address line 4). The fourth address line for the user. This field is optional input to the directory and can contain blanks.
- *WOSDLOC* (Location). The location of the user. This can specify an item such as a site or department. This field is optional input to the directory and can contain blanks.
- *WOSDTEL1* (Telephone number 1). The telephone number of the user. This field is optional input to the directory and can contain blanks.
- *WOSDTEL2* (Telephone number 2). The second or alternative telephone number of the user. This field is optional input to the directory and can contain blanks.
- *WOSDTEXT* (Text). Any additional information about the user can be contained in this field. It is optional input to the directory and can contain blanks.
- *WOSDCS* (System name). The system on which the output file was created.
- *WOSDCC* (Century). The century in which the output file was created (0=1900, 1=2000).
- *WOSDCD* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *WOSDCT* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *WOSDFAX* (Fax telephone number). The fax telephone number for the user. The number can be up to 32 characters long. It is optional input to the directory and can contain blanks.

The following data description specifications were used to create the model output file when `OUTFILFMT(*TYPE2)` and `DETAIL(*BASIC)` or `OUTFILFMT(*TYPE3)` and `DETAIL(*BASIC)` are specified:

```

A* Start of specifications
A*
A* File Name: QAOSDIRB in library QSYS
A* File Type: Physical
A* Record Format Name: OSDIRB
A*
A*
A          R OSDIRB                                TEXT('Directory Entry +
                                                TYPE2 Basic Record
                                                (DSPDIR)')
A          DBDEN          8A          TEXT('User ID')
A          DBDGN          8A          TEXT('Address')
A          DBDESC        50A          TEXT('User Description')
A          DBCSYS        8A          TEXT('System the outfile +
                                                was created on')
A          DBCCEN        1A          TEXT('Century the outfile +
                                                was created')
A          DBCDAT        6A          TEXT('Date the outfile +
                                                was created')
A          DBCTIM        6A          TEXT('Time the outfile +
                                                was created')
A
A*
A* End of specifications

```

Figure C-2. Output File `OUTFILFMT (*TYPE2)` or `(*TYPE3)` and `DETAIL (*BASIC)` Example

The meaning of each field in the record is:

- **DBDEN** (User ID). The user identification for which the detailed information is provided. This field is put into code from code page 500/character set 930. The user ID along with the address uniquely identifies the directory entry.
- **DBDGN** (Address). The address for which the detailed information is provided. This field is put into code from code page 500/character set 930. The address along with the user ID uniquely identifies the directory entry.
- **DBDESC** (User description). The user description associated with the user ID and address.
- **DBCSYS** (System name). The system on which the output file was created.
- **DBCCEN** (Century). The century in which the output file was created (0=1900, 1=2000).
- **DBCDAT** (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- **DBCTIM** (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.

The following data description specifications are used to create the model output file when OUTFILFMT(*TYPE2) and DETAIL(*FULL) or a specific user is specified:

```

A* Start of specifications
A*
A* File Name: QAOSDIRF in library QSYS
A* File Type: Physical
A* Record Format Name: OSDIRF
A*
A*
A      R OSDIRF                TEXT('Directory Entry +
                                TYPE2 Full Record +
                                (DSPDIR)')
A      DF DEN      8A          TEXT('User ID')
A      DF DGN      8A          TEXT('Address')
A      DF DESC     50A         TEXT('User Description')
A      DF REN      8A          TEXT('System Name')
A      DF RGN      8A          TEXT('System Group')
A      DF USRP     10A         TEXT('User Profile')
A      DF INDU     1A          TEXT('Indirect User Flag')
A      DF RCP      1A          TEXT('Print Personal Mail')
A      DF LSTN     40A         TEXT('Last Name')
A      DF LNCS     3S          TEXT('Last Name +
                                Character Set')
A      DF LNCP     3S          TEXT('Last Name Code Page')
A      DF FSTN     20A         TEXT('First Name')
A      DF FNCS     3S          TEXT('First Name +
                                Character Set')
A      DF FNCP     3S          TEXT('First Name Code Page')
A      DF MIDN     20A         TEXT('Middle Name')
A      DF MNCS     3S          TEXT('Middle Name +
                                Character Set')
A      DF MNCP     3S          TEXT('Middle Name Code Page')
A      DF PRFN     20A         TEXT('Preferred Name')
A      DF PNCS     3S          TEXT('Preferred Name +
                                Character Set')
A      DF PNCD     3S          TEXT('Preferred Name +
                                Code Page')
A      DF FULN     64A         TEXT('Full Name')
A      DF FLCS     3S          TEXT('Full Name +
                                Character Set')
A      DF FLCP     3S          TEXT('Full Name Code Page')
A      DF DPT      10A         TEXT('Department')
A      DF DP CS    3S          TEXT('Department +
                                Character Set')
A      DF DP CP    3S          TEXT('Department Code Page')
A      DF JOB      40A         TEXT('Job Title')
A      DF JB CS    3S          TEXT('Job Title +
                                Character Set')
A      DF JB CP    3S          TEXT('Job Title Code Page')
A      DF COMP     50A         TEXT('Company Name')
A      DF COCS     3S          TEXT('Company +
                                Character Set')
A      DF CO CP    3S          TEXT('Company Code Page')
A      DF TEL1     26A         TEXT('Telephone Number 1')
A      DF T1 CS    3S          TEXT('Telephone 1 +
                                Character Set')

```

Figure C-3 (Part 1 of 4). Output File *OUTFILFMT(*TYPE2)* and *DETAIL(*FULL)* Example

| | | | |
|---|--------|-----|--|
| A | DFT1CP | 3S | TEXT('Telephone 1 Code Page') |
| A | DFTEL2 | 26A | TEXT('Telephone Number 2') |
| A | DFT2CS | 3S | TEXT('Telephone 2 + Character Set') |
| A | DFT2CP | 3S | TEXT('Telephone 2 Code Page') |
| A | DFLOC | 40A | TEXT('Location') |
| A | DFLOCS | 3S | TEXT('Location + Character Set') |
| A | DFLOCP | 3S | TEXT('Location Code Page') |
| A | DFBLD | 20A | TEXT('Building') |
| A | DFBDCS | 3S | TEXT('Building + Character Set') |
| A | DFBDP | 3S | TEXT('Building Code Page') |
| A | DFOFC | 16A | TEXT('Office') |
| A | DFOFCS | 3S | TEXT('Office + Character Set') |
| A | DFOFCP | 3S | TEXT('Office Code Page') |
| A | DFADD1 | 40A | TEXT('Mailing Address 1') |
| A | DFA1CS | 3S | TEXT('Address 1 + Character Set') |
| A | DFA1CP | 3S | TEXT('Address 1 Code Page') |
| A | DFADD2 | 40A | TEXT('Mailing Address 2') |
| A | DFA2CS | 3S | TEXT('Address 2 + Character Set') |
| A | DFA2CP | 3S | TEXT('Address 2 Code Page') |
| A | DFADD3 | 40A | TEXT('Mailing Address 3') |
| A | DFA3CS | 3S | TEXT('Address 3 + Character Set') |
| A | DFA3CP | 3S | TEXT('Address 3 Code Page') |
| A | DFADD4 | 40A | TEXT('Mailing Address 4') |
| A | DFA4CS | 3S | TEXT('Address 4 + Character Set') |
| A | DFA4CP | 3S | TEXT('Address 4 Code Page') |
| A | DFTEXT | 50A | TEXT('Miscellaneous TEXT') |
| A | DFTXCS | 3S | TEXT('Miscellaneous TEXT + Character Set') |
| A | DFTXCP | 3S | TEXT('Miscellaneous TEXT + Code Page') |
| A | DFLUdT | 6A | TEXT('Last Update Date') |
| A | DFLUTM | 6A | TEXT('Last Update Time') |
| A | DFLUDE | 8A | TEXT('Last Update User ID') |
| A | DFLUDG | 8A | TEXT('Last Update Address') |
| A | DFCSYS | 8A | TEXT('System the outfile + was created on') |
| A | DFCCEN | 1A | TEXT('Century the outfile + was created') |
| A | DFCDAT | 6A | TEXT('Date the outfile + was created') |
| A | DFCTIM | 6A | TEXT('Time the outfile + was created') |
| A | DFCVRP | 1A | TEXT('Print Cover Page') |
| A | DFMLNT | 1A | TEXT('Mail Notification') |
| A | DFPPML | 1A | TEXT('Priority or + Personal Mail') |
| A | DFMSG | 1A | TEXT('Messages') |

Figure C-3 (Part 2 of 4). Output File OUTFILFMT(*TYPE2) and DETAIL(*FULL) Example

I

| | | | |
|---|--------|-----|---|
| A | DFNUID | 47A | TEXT('Network User ID') |
| A | DFFWDU | 8A | TEXT('Forward from User ID') |
| A | DFFWDA | 8A | TEXT('Forward from Address') |
| A | DFNWUI | 8A | TEXT('Pending New User ID') |
| A | DFNWAD | 8A | TEXT('Pending New Address') |
| A | DFRNMF | 1A | TEXT('Rename Status + blank=Not in process + R=In Process + E=In Error') |
| A | DFLCLI | 1A | TEXT('Local Indicator + 0=Local 1=Shadowed') |
| A | DFLCS4 | 4S | TEXT('Last Name + Character Set') |
| A | DFLCP4 | 4S | TEXT('Last Name + Code Page') |
| A | DFFCS4 | 4S | TEXT('First Name + Character Set') |
| A | DFFCP4 | 4S | TEXT('First Name + Code Page') |
| A | DFMCS4 | 4S | TEXT('Middle Name + Character Set') |
| A | DFMCP4 | 4S | TEXT('Middle Name + Code Page') |
| A | DFPCS4 | 4S | TEXT('Preferred Name + Character Set') |
| A | DFPCP4 | 4S | TEXT('Preferred Name + Code Page') |
| A | DFFLS4 | 4S | TEXT('Full Name + Character Set') |
| A | DFFLP4 | 4S | TEXT('Full Name + Code Page') |
| A | DFDCS4 | 4S | TEXT('Department + Character Set') |
| A | DFDCP4 | 4S | TEXT('Department + Code Page') |
| A | DFJCS4 | 4S | TEXT('Job Title + Character Set') |
| A | DFJCP4 | 4S | TEXT('Job Title + Code Page') |
| A | DFCCS4 | 4S | TEXT('Company + Character Set') |
| A | DFCCP4 | 4S | TEXT('Company + Code Page') |
| A | DF1CS4 | 4S | TEXT('Telephone 1 + Character Set') |
| A | DF1CP4 | 4S | TEXT('Telephone 1 + Code Page') |
| A | DF2CS4 | 4S | TEXT('Telephone 2 + Character Set') |
| A | DF2CP4 | 4S | TEXT('Telephone 2 + Code Page') |
| A | DFLTS4 | 4S | TEXT('Location + Character Set') |
| A | DFLTP4 | 4S | TEXT('Location + Code Page') |

Figure C-3 (Part 3 of 4). Output File *OUTFILFMT(*TYPE2)* and *DETAIL(*FULL)* Example

| | | | |
|---|--------|-----|--|
| A | DFBCS4 | 4S | TEXT('Building + Character Set') |
| A | DFBCP4 | 4S | TEXT('Building + Code Page') |
| A | DFOCS4 | 4S | TEXT('Office + Character Set') |
| A | DFOCP4 | 4S | TEXT('Office + Code Page') |
| A | DFA1S4 | 4S | TEXT('Address 1 + Character Set') |
| A | DFA1P4 | 4S | TEXT('Address 1 + Code Page') |
| A | DFA2S4 | 4S | TEXT('Address 2 + Character Set') |
| A | DFA2P4 | 4S | TEXT('Address 2 + Code Page') |
| A | DFA3S4 | 4S | TEXT('Address 3 + Character Set') |
| A | DFA3P4 | 4S | TEXT('Address 3 + Code Page') |
| A | DFA4S4 | 4S | TEXT('Address 4 + Character Set') |
| A | DFA4P4 | 4S | TEXT('Address 4 + Code Page') |
| A | DFTCS4 | 4S | TEXT('Text + Character Set') |
| A | DFTCP4 | 4S | TEXT('Text + Code Page') |
| A | DFFXNB | 32A | TEXT('Fax telephone + Number') |
| A | DFFXCS | 4S | TEXT('Fax telephone + Character Set') |
| A | DFFXCP | 4S | TEXT('Fax telephone + Code Page') |

A*

A* End of specifications

Figure C-3 (Part 4 of 4). Output File *OUTFILFMT(*TYPE2)* and *DETAIL(*FULL)* Example

The meaning of each field in the record is:

- **DFDEN** (User ID). The user identification for which the detailed information is provided. This field is put into code from code page 500, character set 930. The user ID along with the address uniquely identifies the directory entry.
- **DFDGN** (Address). The address for which the detailed information is provided. This field is put into code from code page 500, character set 930. The address, along with the user ID, uniquely identifies the directory entry.
- **DFDESC** (User description). The user description associated with the user ID and address.
- **DFREN** (System name). The system name (known to SNADS) of the system on which this user works. This field is put into code from code page 500, character set 930. The system name along with the system group uniquely identifies a system.
- **DFRGN** (System group). The system group (known to SNADS) of the system on which this user works. This field is put into code from code page 500, char-

acter set 930. The system group can be blank. The system group, along with the system name, uniquely identifies a system.

- *DFUSRP* (User profile). The system profile name for the user.
- *DFINDU* (Indirect user flag). A 1-character flag to indicate whether or not the user is an indirect user (one who does not use the system to receive mail, but receives printed mail). The value 1 indicates the user is an indirect user. The value 0 indicates that the user is not an indirect user.
- *DFRCVP* (Receive personal mail). A 1-character flag to specify whether an indirect user is to have personal mail printed. The value 1 indicates the indirect user is to receive personal mail. The value 0 indicates that no personal mail is to be printed for the indirect user. The data in this field has no meaning if this is not an indirect user.
- *DFLSTN* (Last name). The last name of the user. This field is optional input to the directory and can contain blanks.
- *DFLNCS* (Last name character set). This is the character set of the field *DFLSTN* (Last name).
- *DFLNCP* (Last name code page). This is the code page of the field *DFLSTN* (Last name).
- *DFFSTN* (First name). The first name of the user. This field is optional input to the directory and can contain blanks.
- *DFFNCS* (First name character set). This is the character set of the field *DFFSTN* (First name).
- *DFFNCP* (First name code page). This is the code page of the field *DFFSTN* (First name).
- *DFMIDN* (Middle name). The middle name of the user. This field is optional input to the directory and can contain blanks.
- *DFMNCS* (Middle name character set). This is the character set of the field *DFMIDN* (Middle name).
- *DFMNCP* (Middle name code page). This is the code page of the field *DFMIDN* (Middle name).
- *DFPRFN* (Preferred name). The preferred name of the user. This field is optional input to the directory and can contain blanks.
- *DFPNCS* (Preferred name character set). This is the character set of the field *DFPRFN* (Preferred name).
- *DFPNCD* (Preferred name code page). This is the code page of the field *DFPRFN* (Preferred name).
- *DFFULN* (Full name). The full name of the user. This field is optional input to the directory and can contain blanks.
- *DFFLCS* (Full name character set). This is the character set of the field *DFFULN* (Full name).
- *DFFLCP* (Full name code page). This is the code page of the field *DFFULN* (Full name).
- *DFDPT* (Department). The department the user is in. This field is optional input to the directory and can contain blanks.

- *DFDPCS* (Department character set). This is the character set of the field DFDPT (Department).
- *DFDPCP* (Department code page). This is the code page of the field DFDPT (Department).
- *DFJOB* (Job title). This is the job title of the user. This field is optional input to the directory and can contain blanks.
- *DFJBCS* (Job title character set). This is the character set of the field DFJOB (Job title).
- *DFJBPCP* (Job title code page). This is the code page of the field DFJOB (Job title).
- *DFCOMP* (Company). This is the company the user is in. This field is optional input to the directory and can contain blanks.
- *DFCOCS* (Company character set). This is the character set of the field DFCOMP (Company).
- *DFCOCP* (Company code page). This is the code page of the field DFCOMP (Company).
- *DFTEL1* (Telephone number 1). The telephone number of the user. This field is optional input to the directory and can contain blanks.
- *DFT1CS* (Telephone number 1 character set). This is the character set of the field DFTEL1 (Telephone number 1).
- *DFT1CP* (Telephone number 1 code page). This is the code page of the field DFTEL1 (Telephone number 1).
- *DFTEL2* (Telephone number 2). The telephone number of the user. This field is optional input to the directory and can contain blanks.
- *DFT2CS* (Telephone number 2 character set). This is the character set of the field DFTEL2 (Telephone number 2).
- *DFT2CP* (Telephone number 2 code page). This is the code page of the field DFTEL2 (Telephone number 2).
- *DFLOC* (Location). The location of the user. This field is optional input to the directory and can contain blanks.
- *DFLOCS* (Location character set). This is the character set of the field DFLOC (Location).
- *DFLOCP* (Location code page). This is the code page of the field DFLOC (Location).
- *DFBLD* (Building). The building in which the user is located. This field is optional input to the directory and can contain blanks.
- *DFBDCS* (Building character set). This is the character set of the field DFBLD (Building).
- *DFBDPCP* (Building code page). This is the code page of the field DFBLD (Building).
- *DFOFC* (Office). The office the user is in. This field is optional input to the directory and can contain blanks.
- *DFOFCS* (Office character set). This is the character set of the field DFOFC (Office).

- *DFOFCP* (Office code page). This is the code page of the field DFOFC (Office).
- *DFADD1* (Mailing address 1). The first address line for the user. This field is optional input to the directory and can contain blanks.
- *DFA1CS* (Address 1 character set). This is the character set of the field DFADD1 (Mailing address 1).
- *DFA1CP* (Address 1 code page). This is the code page of the field DFADD1 (Mailing address 1).
- *DFADD2* (Mailing address 2). The second address line for the user. This field is optional input to the directory and can contain blanks.
- *DFA2CS* (Address 2 character set). This is the character set of the field DFADD2 (Mailing address 2).
- *DFA2CP* (Address 2 code page). This is the code page of the field DFADD2 (Mailing address 2).
- *DFADD3* (Mailing address 3). The third address line for the user. This field is optional input to the directory and can contain blanks.
- *DFA3CS* (Address 3 character set). This is the character set of the field DFADD3 (Mailing address 3).
- *DFA3CP* (Address 3 code page). This is the code page of the field DFADD3 (Mailing address 3).
- *DFADD4* (Mailing address 4). The fourth address line for the user. This field is optional input to the directory and can contain blanks.
- *DFA4CS* (Address 4 character set). This is the character set of the field DFADD4 (Mailing address 4).
- *DFA4CP* (Address 4 code page). This is the code page of the field DFADD4 (Mailing address 4).
- *DFTEXT* (Text). Any additional information about the user can be contained in this field. This field is optional input to the directory and can contain blanks.
- *DFTXCS* (Text character set). This is the character set of the field DFTEXT (Text).
- *DFTXCP* (Text code page). This is the code page of the field DFTEXT (Text).
- *DFLUDT* (Last update date). This field contains the last date on which the directory entry was updated. The date is in the format YYMMDD where YY=year, MM=month, and DD=day.
- *DFLUTM* (Last update time). This field contains the last time that the directory entry was updated. The time is in the format HHMMSS where HH=hour, MM=minutes, and SS=seconds.
- *DFLUDE* (Last update user ID). This field contains the user identification of the user who last updated the directory entry.
- *DFLUDG* (Last update address). This field contains the address of the user who last updated the directory entry.
- *DFCSYS* (System name). The system on which the output file was created.
- *DFCCEN* (Century). The century in which the output file was created (0=1900, 1=2000).

- *DFCDAT* (Date). The date the output file was created. The date is specified as YYMMDD where YY=year, MM=month, and DD=day.
- *DFCTIM* (Time). The time the output file was created. The date is specified as HHMMSS where HH=hour, MM=minutes, and SS=seconds.
- *DFCVRP* (Print cover page). This field contains a 1-character flag to specify whether a direct or indirect user desires a cover page for printed mail (Y=Yes, N=No).
- *DFMLNT* (Mail notification). This field contains a 1-character flag to specify whether a direct or indirect user is notified of the arrival of all mail, certain types of mail, or no mail (1=Specific types of mail, 2=All mail, 3=No mail).
- *DFPPML* (Priority or personal mail). This field contains a 1-character flag to specify whether a direct or indirect user is notified of the arrival of priority or personal mail (Y=Yes, N=No).
- *DFMSG* (Messages). This field contains a 1-character flag to specify whether a direct or indirect user is notified of the arrival of messages (Y=Yes, N=No).
- *DFNUID* (Network user ID). This field contains the network user ID of the user.
- *DFFWDU* (Forward from user ID). This field contains the user ID from which distributions are automatically forwarded. The field is put into code from character set 930 and code page 500. The user ID, along with the address, identify who distributions should be forwarded from.
- *DFFWDA* (Forward from address). This field contains the address from which distributions are automatically forwarded. The field is put into code from character set 930 and code page 500. The address, along with the user ID, identify who distributions should be forwarded from.
- *DFNWUI* (Pending new user ID). This field contains the new user ID if the user is being renamed. The field is put into code from character set 930 and code page 500. This field only contains a value if the user is in the process of being renamed or the rename failed.
- *DFNWAD* (Pending new address). This field contains the new address if the user is being renamed. The field is put into code from character set 930 and code page 500. This field only contains a value if the user is in the process of being renamed or the rename failed.
- *DFRNMF* (Rename status). This field contains a 1-character flag to specify the rename status of the user (Blank=rename not in process, R=rename in process, and E=rename ended in error).
- *DFLCLI* (Local indicator). This field contains a 1-character flag to specify if the user originated on this system (0=local, originated on this system; 1=shadowed, supplied from another system).
- *DFLCS4* (Last name character set). This is the character set of the field DFLSTN (Last name).
- *DFLCP4* (Last name code page). This is the code page of the field DFLSTN (Last name).
- *DFFC4* (First name character set). This is the character set of the field DFFSTN (First name).
- *DFFCP4* (First name code page). This is the code page of the field DFFSTN (First name).

- *DFMCS4* (Middle name character set). This is the character set of the field DF MIDN (Middle name).
- *DFMCP4* (Middle name code page). This is the code page of the field DF MIDN (Middle name).
- *DFPCS4* (Preferred name character set). This is the character set of the field DF PRFN (Preferred name).
- *DFPCP4* (Preferred name code page). This is the code page of the field DF PRFN (Preferred name).
- *DFFLS4* (Full name character set). This is the character set of the field DF FULN (Full name).
- *DFFLP4* (Full name code page). This is the code page of the field DF FULN (Full name).
- *DFDCS4* (Department character set). This is the character set of the field DF DPT (Department).
- *DFDCP4* (Department code page). This is the code page of the field DF DPT (Department).
- *DFJCS4* (Job title character set). This is the character set of the field DF JOB (Job title).
- *DFJCP4* (Job title code page). This is the code page of the field DF JOB (Job title).
- *DFCCS4* (Company character set). This is the character set of the field DF COMP (Company).
- *DFCCP4* (Company code page). This is the code page of the field DF COMP (Company).
- *DF1CS4* (Telephone number 1 character set). This is the character set of the field DF TEL1 (Telephone number 1).
- *DF1CP4* (Telephone number 1 code page). This is the code page of the field DF TEL1 (Telephone number 1).
- *DF2CS4* (Telephone number 2 character set). This is the character set of the field DF TEL2 (Telephone number 2).
- *DF2CP4* (Telephone number 2 code page). This is the code page of the field DF TEL2 (Telephone number 2).
- *DFLTS4* (Location character set). This is the character set of the field DF LOC (Location).
- *DFLTP4* (Location code page). This is the code page of the field DF LOC (Location).
- *DFBCS4* (Building character set). This is the character set of the field DF BLD (Building).
- *DFBCP4* (Building code page). This is the code page of the field DF BLD (Building).
- *DFOCS4* (Office character set). This is the character set of the field DF OFC (Office).
- *DFOCP4* (Office code page). This is the code page of the field DF OFC (Office).

- *DFA1S4* (Address 1 character set). This is the character set of the field DFADD1 (Mailing address 1).
- *DFA1P4* (Address 1 code page). This is the code page of the field DFADD1 (Mailing address 1).
- *DFA2S4* (Address 2 character set). This is the character set of the field DFADD2 (Mailing address 2).
- *DFA2P4* (Address 2 code page). This is the code page of the field DFADD2 (Mailing address 2).
- *DFA3S4* (Address 3 character set). This is the character set of the field DFADD3 (Mailing address 3).
- *DFA3P4* (Address 3 code page). This is the code page of the field DFADD3 (Mailing address 3).
- *DFA4S4* (Address 4 character set). This is the character set of the field DFADD4 (Mailing address 4).
- *DFA4P4* (Address 4 code page). This is the code page of the field DFADD4 (Mailing address 4).
- *DFTCS4* (Text character set). This is the character set of the field DFTEXT (Text).
- *DFTCP4* (Text code page). This is the code page of the field DFTEXT (Text).
- *DFFXNB* (Fax telephone number). The fax telephone number. This field is optional input to the directory and can contain blanks.
- *DFFXCS* (Fax telephone character set). This is the character set of the field DFFXNB (Fax telephone number).
- *DFFXCP* (Fax telephone code page). This is the code page of the field DFFXNB (Fax telephone number).

The following data description specifications are used to create the model output file when OUTFILFMT(*TYPE3) and DETAIL(*FULL) are specified:

```

A* Start of specifications
A*
A* File Name: QAOSDIRX in library QSYS
A* File Type: Physical
A* Record Format Name: OSDIRX
A*
A*
A          R OSDIRX          TEXT('Directory Entry 0/R +
                                Names (DSPDIR)')
A          DXDEN            8A    TEXT('User ID')
A          DXDGN            8A    TEXT('Address')
A          DXDESC           50A    TEXT('User Description')
A          DXREN            8A    TEXT('System Name')
A          DXRGN            8A    TEXT('System Group')
A          DXUSRP           10A    TEXT('User Profile')
A          DXINDU            1A    TEXT('Indirect User Flag')
A          DXRCVP            1A    TEXT('Print Personal Mail')
A          DXLSTN           40A    TEXT('Last Name')
A          DXLNCS            3S    TEXT('Last Name +
                                Character Set')
A          DXLNCP            3S    TEXT('Last Name Code Page')
A          DXFSTN           20A    TEXT('First Name')
A          DXFNCS            3S    TEXT('First Name +
                                Character Set')
A          DXFNCP            3S    TEXT('First Name Code Page')
A          DXMIDN           20A    TEXT('Middle Name')
A          DXMNCS            3S    TEXT('Middle Name +
                                Character Set')
A          DXMNCP            3S    TEXT('Middle Name Code Page')
A          DXPRFN           20A    TEXT('Preferred Name')
A          DXPNCS            3S    TEXT('Preferred Name +
                                Character Set')
A          DXPNCD            3S    TEXT('Preferred Name +
                                Code Page')
A          DXFULN           64A    TEXT('Full Name')
A          DXFLCS            3S    TEXT('Full Name +
                                Character Set')
A          DXFLCP            3S    TEXT('Full Name Code Page')
A          DXDPT            10A    TEXT('Department')
A          DXDPCS            3S    TEXT('Department +
                                Character Set')
A          DXDPCP            3S    TEXT('Department Code Page')
A          DXJOB            40A    TEXT('Job Title')
A          DXJBCS            3S    TEXT('Job Title +
                                Character Set')
A          DXJBCP            3S    TEXT('Job Title Code Page')
A          DXCOMP           50A    TEXT('Company Name')
A          DXCOCS            3S    TEXT('Company +
                                Character Set')
A          DXCOCP            3S    TEXT('Company Code Page')
A          DXTEL1           26A    TEXT('Telephone Number 1')
A          DXT1CS            3S    TEXT('Telephone 1 +
                                Character Set')
A          DXT1CP            3S    TEXT('Telephone 1 Code Page')

```

Figure C-4 (Part 1 of 5). Output File *OUTFILFMT(*TYPE3)* and *DETAIL(*FULL)* Example

| | | | |
|---|---------|-----|---|
| A | DXTEL2 | 26A | TEXT('Telephone Number 2') |
| A | DXT2CS | 3S | TEXT('Telephone 2 + Character Set') |
| A | DXT2CP | 3S | TEXT('Telephone 2 Code Page') |
| A | DXLOC | 40A | TEXT('Location') |
| A | DXLOCS | 3S | TEXT('Location + Character Set') |
| A | DXLOCP | 3S | TEXT('Location Code Page') |
| A | DXBLD | 20A | TEXT('Building') |
| A | DXBDCS | 3S | TEXT('Building + Character Set') |
| A | DXBDCP | 3S | TEXT('Building Code Page') |
| A | DXOFC | 16A | TEXT('Office') |
| A | DXOFCS | 3S | TEXT('Office + Character Set') |
| A | DXOFCP | 3S | TEXT('Office Code Page') |
| A | DXADD1 | 40A | TEXT('Mailing Address 1') |
| A | DXA1CS | 3S | TEXT('Address 1 + Character Set') |
| A | DXA1CP | 3S | TEXT('Address 1 Code Page') |
| A | DXADD2 | 40A | TEXT('Mailing Address 2') |
| A | DXA2CS | 3S | TEXT('Address 2 + Character Set') |
| A | DXA2CP | 3S | TEXT('Address 2 Code Page') |
| A | DXADD3 | 40A | TEXT('Mailing Address 3') |
| A | DXA3CS | 3S | TEXT('Address 3 + Character Set') |
| A | DXA3CP | 3S | TEXT('Address 3 Code Page') |
| A | DXADD4 | 40A | TEXT('Mailing Address 4') |
| A | DXA4CS | 3S | TEXT('Address 4 + Character Set') |
| A | DXA4CP | 3S | TEXT('Address 4 Code Page') |
| A | DXTEXT | 50A | TEXT('Miscellaneous TEXT') |
| A | DXTXCS | 3S | TEXT('Miscellaneous TEXT + Character Set') |
| A | DXTXCP | 3S | TEXT('Miscellaneous TEXT + Code Page') |
| A | DXLUDT | 6A | TEXT('Last Update Date') |
| A | DXLUTM | 6A | TEXT('Last Update Time') |
| A | DXLUDE | 8A | TEXT('Last Update User ID') |
| A | DXLUDG | 8A | TEXT('Last Update Address') |
| A | DXCVRP | 1A | TEXT('Print Cover Page') |
| A | DXMLNT | 1A | TEXT('Mail Notification') |
| A | DXPPML | 1A | TEXT('Priority or + Personal Mail') |
| A | DXMSGGS | 1A | TEXT('Messages') |
| A | DXCTRY | 3A | TEXT('Country') |
| A | DXADMN | 16A | TEXT('Administrative Domain') |
| A | DXPDMN | 16A | TEXT('Private Domain') |
| A | DXORG | 64A | TEXT('Organization') |

Figure C-4 (Part 2 of 5). Output File *OUTFILFMT(*TYPE3)* and *DETAIL(*FULL)* Example

| | | | |
|---|--------|------|---|
| A | DXPNSN | 40A | TEXT('Personal Name - + Surname') |
| A | DXRSV1 | 24A | TEXT('Reserved') |
| A | DXPNGN | 16A | TEXT('Personal Name - + Given name') |
| A | DXRSV2 | 4A | TEXT('Reserved') |
| A | DXPNIN | 5A | TEXT('Personal Name - + Initials') |
| A | DXPNGQ | 3A | TEXT('Personal Name - + Generation Qualifier') |
| A | DXRSV3 | 7A | TEXT('Reserved') |
| A | DXOU1 | 32A | TEXT('Organization Unit 1') |
| A | DXOU2 | 32A | TEXT('Organization Unit 2') |
| A | DXOU3 | 32A | TEXT('Organization Unit 3') |
| A | DXOU4 | 32A | TEXT('Organization Unit 4') |
| A | DXDAT1 | 8A | TEXT('Domain-defined ' + 'Attribute Type 1') |
| A | DXDAV1 | 128A | TEXT('Domain-defined ' + 'Attribute Value 1') |
| A | DXDAT2 | 8A | TEXT('Domain-defined ' + 'Attribute Type 2') |
| A | DXDAV2 | 128A | TEXT('Domain-defined ' + 'Attribute Value 2') |
| A | DXDAT3 | 8A | TEXT('Domain-defined ' + 'Attribute Type 3') |
| A | DXDAV3 | 128A | TEXT('Domain-defined ' + 'Attribute Value 3') |
| A | DXDAT4 | 8A | TEXT('Domain-defined ' + 'Attribute Type 4') |
| A | DXDAV4 | 128A | TEXT('Domain-defined ' + 'Attribute Value 4') |
| A | DXCSYS | 8A | TEXT('System the outfile + was created on') |
| A | DXCCEN | 1A | TEXT('Century the outfile + was created') |
| A | DXCDAT | 6A | TEXT('Date the outfile + was created') |
| A | DXCTIM | 6A | TEXT('Time the outfile + was created') |
| A | DXNUID | 47A | TEXT('Network User ID') |
| A | DXFWDU | 8A | TEXT('Forward from User ID') |
| A | DXFWDA | 8A | TEXT('Forward from Address') |
| A | DXNWUI | 8A | TEXT('Pending New User ID') |
| A | DXNWAD | 8A | TEXT('Pending New Address') |
| A | DXRNMF | 1A | TEXT('Rename Status + blank=Not in process + R=In Process + E=In Error') |
| A | DXLCLI | 1A | TEXT('Local Indicator + 0=Local 1=Shadowed') |

Figure C-4 (Part 3 of 5). Output File *OUTFILFMT(*TYPE3)* and *DETAIL(*FULL)* Example

| | | | |
|---|--------|----|---|
| A | DXLCS4 | 4S | TEXT('Last Name + Character Set') |
| A | DXLCP4 | 4S | TEXT('Last Name + Code Page') |
| A | DXFCS4 | 4S | TEXT('First Name + Character Set') |
| A | DXFCP4 | 4S | TEXT('First Name + Code Page') |
| A | DXMCS4 | 4S | TEXT('Middle Name + Character Set') |
| A | DXMCP4 | 4S | TEXT('Middle Name + Code Page') |
| A | DXPCS4 | 4S | TEXT('Preferred Name + Character Set') |
| A | DXPCP4 | 4S | TEXT('Preferred Name + Code Page') |
| A | DXFLS4 | 4S | TEXT('Full Name + Character Set') |
| A | DXFLP4 | 4S | TEXT('Full Name + Code Page') |
| A | DXDCS4 | 4S | TEXT('Department + Character Set') |
| A | DXDCP4 | 4S | TEXT('Department + Code Page') |
| A | DXJCS4 | 4S | TEXT('Job Title + Character Set') |
| A | DXJCP4 | 4S | TEXT('Job Title + Code Page') |
| A | DXCCS4 | 4S | TEXT('Company + Character Set') |
| A | DXCCP4 | 4S | TEXT('Company + Code Page') |
| A | DX1CS4 | 4S | TEXT('Telephone 1 + Character Set') |
| A | DX1CP4 | 4S | TEXT('Telephone 1 + Code Page') |
| A | DX2CS4 | 4S | TEXT('Telephone 2 + Character Set') |
| A | DX2CP4 | 4S | TEXT('Telephone 2 + Code Page') |
| A | DXLTS4 | 4S | TEXT('Location + Character Set') |
| A | DXLTP4 | 4S | TEXT('Location + Code Page') |
| A | DXBCS4 | 4S | TEXT('Building + Character Set') |
| A | DXBCP4 | 4S | TEXT('Building + Code Page') |
| A | DXOCS4 | 4S | TEXT('Office + Character Set') |
| A | DXOCP4 | 4S | TEXT('Office + Code Page') |
| A | DXA1S4 | 4S | TEXT('Address 1 + Character Set') |
| A | DXA1P4 | 4S | TEXT('Address 1 + Code Page') |

Figure C-4 (Part 4 of 5). Output File *OUTFILFMT(*TYPE3)* and *DETAIL(*FULL)* Example

| | | | |
|---|--------|-----|--|
| A | DXA2S4 | 4S | TEXT('Address 2 + Character Set') |
| A | DXA2P4 | 4S | TEXT('Address 2 + Code Page') |
| A | DXA3S4 | 4S | TEXT('Address 3 + Character Set') |
| A | DXA3P4 | 4S | TEXT('Address 3 + Code Page') |
| A | DXA4S4 | 4S | TEXT('Address 4 + Character Set') |
| A | DXA4P4 | 4S | TEXT('Address 4 + Code Page') |
| A | DXTCS4 | 4S | TEXT('Text + Character Set') |
| A | DXTCP4 | 4S | TEXT('Text + Code Page') |
| A | DXFXNB | 32A | TEXT('Fax telephone + number') |
| A | DXFXCS | 4S | TEXT('Fax telephone + Character Set') |
| A | DXFXCP | 4S | TEXT('Fax telephone + Code Page') |

A*
A* End of specifications

Figure C-4 (Part 5 of 5). Output File *OUTFILFMT(*TYPE3)* and *DETAIL(*FULL)* Example

The meaning of each field in the record is:

- *DXDEN* (User ID). The user identification for which the detailed information is provided. This field is put into code from code page 500, character set 930. The user ID, along with the address, uniquely identifies the directory entry.
- *DXDGN* (Address). The address for which the detailed information is provided. This field is put into code from code page 500, character set 930. The address, along with the user ID, uniquely identifies the directory entry.
- *DXDESC* (User description). The user description associated with the user ID and address.
- *DXREN* (System name). The system name (known to SNADS) of the system on which this user works. This field is put into code from code page 500, character set 930. The system name, along with the system group, uniquely identifies a system.
- *DXRGN* (System group). The system group (known to SNADS) of the system on which this user works. This field is put into code from code page 500, character set 930. The system group can be blank. The system group, along with the system name, uniquely identifies a system.
- *DXUSRP* (User profile). The system profile name for the user.
- *DXINDU* (Indirect user flag). A 1-character flag to indicate whether or not the user is an indirect user (one who does not use the system to receive mail, but receives printed mail). The value 1 indicates the user is an indirect user. The value 0 indicates that the user is not an indirect user.
- *DXRCVP* (Receive personal mail). A 1-character flag to specify whether an indirect user is to have personal mail printed. The value 1 indicates the indirect user is to receive personal mail. The value 0 indicates that no personal mail is

to be printed for the indirect user. The data in this field has no meaning if this is not an indirect user.

- *DXLSTN* (Last name). The last name of the user. This field is optional input to the directory and can contain blanks.
- *DXLNCS* (Last name character set). This is the character set of the field *DXLSTN* (Last name).
- *DXLNCP* (Last name code page). This is the code page of the field *DXLSTN* (Last name).
- *DXFSTN* (First name). The first name of the user. This field is optional input to the directory and can contain blanks.
- *DXFNCS* (First name character set). This is the character set of the field *DXFSTN* (First name).
- *DXFNCP* (First name code page). This is the code page of the field *DXFSTN* (First name).
- *DXMIDN* (Middle name). The middle name of the user. This field is optional input to the directory and can contain blanks.
- *DXMNCS* (Middle name character set). This is the character set of the field *DXMIDN* (Middle name).
- *DXMNCP* (Middle name code page). This is the code page of the field *DXMIDN* (Middle name).
- *DXPRFN* (Preferred name). The preferred name of the user. This field is optional input to the directory and can contain blanks.
- *DXPNCS* (Preferred name character set). This is the character set of the field *DXPRFN* (Preferred name).
- *DXPNCD* (Preferred name code page). This is the code page of the field *DXPRFN* (Preferred name).
- *DXFULN* (Full name). The full name of the user. This field is optional input to the directory and can contain blanks.
- *DXFLCS* (Full name character set). This is the character set of the field *DXFULN* (Full name).
- *DXFLCP* (Full name code page). This is the code page of the field *DXFULN* (Full name).
- *DXDPT* (Department). The department the user is in. This field is optional input to the directory and can contain blanks.
- *DXDPCS* (Department character set). This is the character set of the field *DXDPT* (Department).
- *DXDPCP* (Department code page). This is the code page of the field *DXDPT* (Department).
- *DXJOB* (Job title). This is the job title of the user. This field is optional input to the directory and can contain blanks.
- *DXJBOS* (Job title character set). This is the character set of the field *DXJOB* (Job title).
- *DXJBOS* (Job title code page). This is the code page of the field *DXJOB* (Job title).

- *DXCOMP* (Company Name). This is the company the user is in. This field is optional input to the directory and can contain blanks.
- *DXCOCS* (Company character set). This is the character set of the field *DXCOMP* (Company Name).
- *DXCOCP* (Company code page). This is the code page of the field *DXCOMP* (Company Name).
- *DXTEL1* (Telephone number 1). The telephone number of the user. This field is optional input to the directory and can contain blanks.
- *DXT1CS* (Telephone number 1 character set). This is the character set of the field *DXTEL1* (Telephone number 1).
- *DXT1CP* (Telephone number 1 code page). This is the code page of the field *DXTEL1* (Telephone number 1).
- *DXTEL2* (Telephone number 2). The telephone number of the user. This field is optional input to the directory and can contain blanks.
- *DXT2CS* (Telephone number 2 character set). This is the character set of the field *DXTEL2* (Telephone number 2).
- *DXT2CP* (Telephone number 2 code page). This is the code page of the field *DXTEL2* (Telephone number 2).
- *DXLOC* (Location). The location of the user. This field is optional input to the directory and can contain blanks.
- *DXLOCS* (Location character set). This is the character set of the field *DXLOC* (Location).
- *DXLOCP* (Location code page). This is the code page of the field *DXLOC* (Location).
- *DXBLD* (Building). The building in which the user is located. This field is optional input to the directory and can contain blanks.
- *DXBDCS* (Building character set). This is the character set of the field *DXBLD* (Building).
- *DXBDP* (Building code page). This is the code page of the field *DXBLD* (Building).
- *DXOFC* (Office). The office the user is in. This field is optional input to the directory and can contain blanks.
- *DXOFCS* (Office character set). This is the character set of the field *DXOFC* (Office).
- *DXOFCP* (Office code page). This is the code page of the field *DXOFC* (Office).
- *DXADD1* (Mailing address 1). The first address line for the user. This field is optional input to the directory and can contain blanks.
- *DXA1CS* (Address 1 character set). This is the character set of the field *DXADD1* (Mailing address 1).
- *DXA1CP* (Address 1 code page). This is the code page of the field *DXADD1* (Mailing address 1).
- *DXADD2* (Mailing address 2). The second address line for the user. This field is optional input to the directory and can contain blanks.

- *DXA2CS* (Address 2 character set). This is the character set of the field DXADD2 (Mailing address 2).
- *DXA2CP* (Address 2 code page). This is the code page of the field DXADD2 (Mailing address 2).
- *DXADD3* (Mailing address 3). The third address line for the user. This field is optional input to the directory and can contain blanks.
- *DXA3CS* (Address 3 character set). This is the character set of the field DXADD3 (Mailing address 3).
- *DXA3CP* (Address 3 code page). This is the code page of the field DXADD3 (Mailing address 3).
- *DXADD4* (Mailing address 4). The fourth address line for the user. This field is optional input to the directory and can contain blanks.
- *DXA4CS* (Address 4 character set). This is the character set of the field DXADD4 (Mailing address 4).
- *DXA4CP* (Address 4 code page). This is the code page of the field DXADD4 (Mailing address 4).
- *DXTEXT* (Text). Any additional information about the user can be contained in this field. This field is optional input to the directory and can contain blanks.
- *DXTXCS* (Text character set). This is the character set of the field DXTEXT (Text).
- *DXTXCP* (Text code page). This is the code page of the field DXTEXT (Text).
- *DXLUDT* (Last update date). This field contains the last date on which the directory entry was updated. The date is in the format YYMMDD where YY=year, MM=month, and DD=day.
- *DXLUTM* (Last update time). This field contains the last time that the directory entry was updated. The time is in the format HHMMSS where HH=hour, MM=minutes, and SS=seconds.
- *DXLUDE* (Last update user ID). This field contains the user identification of the user who last updated the directory entry.
- *DXLUDG* (Last update address). This field contains the address of the user who last updated the directory entry.
- *DXCVRP* (Print cover page). This field contains a 1-character flag to specify if either a direct or indirect user desires a cover page for printed mail (Y=Yes, N=No).
- *DXMLNT* (Mail notification). This field contains a 1-character flag to specify if either a direct or indirect user is notified of the arrival of all mail, certain types of mail, or no mail (1=Specific types of mail, 2=All mail, 3=No mail).
- *DXPPML* (Priority or personal mail). This field contains a 1-character flag to specify if either a direct or indirect user is notified of the arrival of priority or personal mail (Y=Yes, N=No).
- *DXMSGs* (Messages). This field contains a 1-character flag to specify if either a direct or indirect user is notified of the arrival of messages (Y=Yes, N=No).
- *DXCTRY* (Country). This field contains the country of the originator/recipient name (O/R name).

- *DXADMN* (Administrative domain). This field contains the administrative management domain of the O/R name. An administrative management domain is a public organization that handles an arrangement domain. A management domain is a set of message transfer agents (MTAs) and user agents (UAs) that comprise a system capable of handling messages.
- *DXPDMN* (Private domain). This field contains the private management domain of the O/R name. A private management domain is a private company or noncommercial organization that handles a management domain. A management domain is a set of message transfer agents (MTAs) and user agents (UAs) that comprise a system capable of handling messages.
- *DXORG* (Organization). This field contains the organization of the O/R name.
- *DXPNSN* (Personal name, surname). This field contains the surname (last name) part of the personal name within the O/R name.
- *DXRSV1* (Reserved). This field is reserved.
- *DXPNGN* (Personal name, given name). This field contains the given name (first name) part of the personal name within the O/R name.
- *DSRSV2* (Reserved). This field is reserved.
- *DXPNIN* (Personal name, initials). This field contains the initials part of the personal name within the O/R name.
- *DXPNGQ* (Personal name, general qualifier). This field contains the generation qualifier part of the personal name within the O/R name.
- *DXRSV3* (Reserved). This field is reserved.
- *DXOU1* (Organization unit 1). This field contains the first organizational unit part of the O/R name.
- *DXOU2* (Organization unit 2). This field contains the second organizational unit part of the O/R name.
- *DXOU3* (Organization unit 3). This field contains the third organizational unit part of the O/R name.
- *DXOU4* (Organization unit 4). This field contains the fourth organizational unit part of the O/R name.
- *DXDAT1* (Domain-defined attribute type 1). This field contains the first domain-defined attribute type part of the O/R name.
- *DXDAV1* (Domain-defined attribute value 1). This field contains the first domain-defined attribute value part of the O/R name.
- *DXDAT2* (Domain-defined attribute type 2). This field contains the second domain-defined attribute type part of the O/R name.
- *DXDAV2* (Domain-defined attribute value 2). This field contains the second domain-defined attribute value part of the O/R name.
- *DXDAT3* (Domain-defined attribute type 3). This field contains the third domain-defined attribute type part of the O/R name.
- *DXDAV3* (Domain-defined attribute value 3). This field contains the third domain-defined attribute value part of the O/R name.
- *DXDAT4* (Domain-defined attribute type 4). This field contains the fourth domain-defined attribute type part of the O/R name.

- *DXDAV4* (Domain-defined attribute value 4). This field contains the fourth-domain defined attribute value part of the O/R name.
- *DXCSYS* (System name). The system on which the output file was created.
- *DXCCEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *DXCDAT* (Date). The date the output file was created. The date is specified as YYMMDD where YY=year, MM=month, and DD=day.
- *DXCTIM* (Time). The time the output file was created. The date is specified as HHMMSS where HH=hour, MM=minutes, and SS=seconds.
- *DXNUID* (Network user ID). This field contains the network user ID of the user.
- *DXFWDU* (Forward from user ID). This field contains the user ID from which distributions are automatically forwarded. The field is put into code from character set 930 and code page 500. The user ID, along with the address, identify who distributions should be forwarded from.
- *DXFWDA* (Forward from address). This field contains the address from which distributions are automatically forwarded. The field is put into code from character set 930 and code page 500. The address, along with the user ID, identify who distributions should be forwarded from.
- *DXNWUI* (Pending new user ID). This field contains the new user ID if the user is being renamed. The field is put into code from character set 930 and code page 500. This field only contains a value if the user is in the process of being renamed or the rename failed.
- *DXNWAD* (Pending new address). This field contains the new address if the user is being renamed. The field is put into code from character set 930 and code page 500. This field only contains a value if the user is in the process of being renamed or the rename failed.
- *DXRNMF* (Rename status). This field contains a 1-character flag to specify the rename status of the user (Blank=rename not in process, R=rename in process, and E=rename ended in error).
- *DXLCLI* (Local indicator). This field contains a 1-character flag to specify if the user originated on this system (0=local, originated on this system; 1=shadowed, supplied from another system).
- *DXLCS4* (Last name character set). This is the character set of the field DXLSTN (Last name).
- *DXLCP4* (Last name code page). This is the code page of the field DXLSTN (Last name).
- *DXFCS4* (First name character set). This is the character set of the field DXFSTN (First name).
- *DXFCP4* (First name code page). This is the code page of the field DXFSTN (First name).
- *DXMCS4* (Middle name character set). This is the character set of the field DXMIDN (Middle name).
- *DXMCP4* (Middle name code page). This is the code page of the field DXMIDN (Middle name).
- *DXPCS4* (Preferred name character set). This is the character set of the field DXPRFN (Preferred name).

- *DXPCP4* (Preferred name code page). This is the code page of the field DXPRFN (Preferred name).
- *DXFLS4* (Full name character set). This is the character set of the field DXFULN (Full name).
- *DXFLP4* (Full name code page). This is the code page of the field DXFULN (Full name).
- *DXDCS4* (Department character set). This is the character set of the field DXDPT (Department).
- *DXDCP4* (Department code page). This is the code page of the field DXDPT (Department).
- *DXJCS4* (Job title character set). This is the character set of the field DXJOB (Job title).
- *DXJCP4* (Job title code page). This is the code page of the field DXJOB (Job title).
- *DXCCS4* (Company character set). This is the character set of the field DXCOMP (Company Name).
- *DXCCP4* (Company code page). This is the code page of the field DXCOMP (Company Name).
- *DX1CS4* (Telephone number 1 character set). This is the character set of the field DXTEL1 (Telephone number 1).
- *DX1CP4* (Telephone number 1 code page). This is the code page of the field DXTEL1 (Telephone number 1).
- *DX2CS4* (Telephone number 2 character set). This is the character set of the field DXTEL2 (Telephone number 2).
- *DX2CP4* (Telephone number 2 code page). This is the code page of the field DXTEL2 (Telephone number 2).
- *DXLTS4* (Location character set). This is the character set of the field DFLOC (Location).
- *DXLTP4* (Location code page). This is the code page of the field DXLOC (Location).
- *DXBCS4* (Building character set). This is the character set of the field DXBLD (Building).
- *DXBCP4* (Building code page). This is the code page of the field DXBLD (Building).
- *DXOCS4* (Office character set). This is the character set of the field DXOFC (Office).
- *DXOCP4* (Office code page). This is the code page of the field DXOFC (Office).
- *DXA1S4* (Address 1 character set). This is the character set of the field DXADD1 (Mailing address 1).
- *DXA1P4* (Address 1 code page). This is the code page of the field DXADD1 (Mailing address 1).
- *DXA2S4* (Address 2 character set). This is the character set of the field DXADD2 (Mailing address 2).

- *DXA2P4* (Address 2 code page). This is the code page of the field DXADD2 (Mailing address 2).
- *DXA3S4* (Address 3 character set). This is the character set of the field DXADD3 (Mailing address 3).
- *DXA3P4* (Address 3 code page). This is the code page of the field DXADD3 (Mailing address 3).
- *DXA4S4* (Address 4 character set). This is the character set of the field DXADD4 (Mailing address 4).
- *DXA4P4* (Address 4 code page). This is the code page of the field DXADD4 (Mailing address 4).
- *DXTCS4* (Text character set). This is the character set of the field DXTEXT (Text).
- *DXTCP4* (Text code page). This is the code page of the field DXTEXT (Text).
- *DXFXNB* (Fax telephone number). This field contains the fax telephone number. This field is optional input to the directory and can contain blanks.
- *DXFXCS* (Fax telephone character set). This is the character set of the field DXFXNB (Fax telephone number).
- *DXFXCP* (Fax telephone code page). This is the code page of the field DXFXNB (Fax telephone number).

Display Document Library Object Auditing Level (DSPDLOAUD) Output File Considerations

The following is how the DSPDLOAUD command produces an output file containing the audit data.

```

* File name = QADSPAUD
A*
A* DSPDLOAUD COMMAND OUTFILE RECORD FORMAT
A*
A          CCSID(65535)
A          R DLOAUD          TEXT('Document Library +
                                Object Detail +
                                (DSPDLOAUD)')
A          CRTCEN          01A          TEXT('Century the outfile +
                                was created')
A          CRTDAT          06A          TEXT('Date the outfile +
                                was created')
A          CRTTIM          06A          TEXT('Time the outfile +
                                was created')
A          SYSNAM          08A          TEXT('System the outfile +
                                was created on')
A          FLRPTH          63A          TEXT('Folder Path')
A          DLONAM          12A          TEXT('DLO Name')
A          OBJNAM          10A          TEXT('System Object Name')
A          DLOTYP          04A          TEXT('DLO Type')
A          AUDLVL          08A          TEXT('Auditing Level')

```

Figure C-5. DSPDLOAUD Command Outfile Record Format

The meaning of each field in the record is:

- *CRTCEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *CRTDAT* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *CRTTIM* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *SYSNAM* (System name). The system on which the output file was created.
- *FLRPTH* (Folder Path). The folder path name, if any, in which the document or folder resides.
- *DLONAM* (Document object library name). The name of the document or folder.
- *OBJNAM* (System Object Name). The system defined name for the document or folder.
- *DLOTYP* (Document Library Object Type). The type of document library object. This may be *FLR or *DOC.
- *AUDLVL* (Auditing Level). The level of auditing active for the specified document or folder.

Display Distribution List (DSPDSTL) Output File Considerations

The Display Distribution List (DSPDSTL) command can be used to create a database file that contains the information about distribution lists. The database file contents are determined by the LSTID and DETAIL parameters.

If the LSTID parameter specified a list ID, the entries within that list are written to the database file. If the LSTID parameter specified *ALL and the DETAIL parameter specified *BASIC, each list ID and description within the directory is written to the database file. When LSTID(*ALL) is specified and DETAIL(*FULL) is specified, each entry within each list is written to the database file.

The database file will contain a header record for each distribution list being written. This header record contains the list ID and description for the printed list. The header record can be followed by detail records containing the user ID/address and description for each of the entries contained in the distribution list.

When the LSTID parameter specified a list ID, a header record is written for the list ID and description. It will be followed by a detail record for each entry in the list.

When the LSTID parameter specified or defaulted to *ALL and the DETAIL parameter specified *FULL, a header record is written for the first distribution list contained in the directory, followed by detail records for all entries in the list. A header record is then written for the next distribution list, again followed by detail records for all entries in that list. This continues until all distribution lists have been processed.

When the LSTID parameter specified or defaulted to *ALL and the DETAIL parameter specified *BASIC, only a header record is written for each distribution list contained in the directory.

The model output file supplied with the system is named QAOSDSTO in library QSYS.

The following data description specifications were used to create the model output file:

```

A* Start of specifications
A*
A* File Name: QAOSDSTO in library QSYS
A* File Type: Physical
A* Record Format Name: OSDSTL
A*
A*
A          R OSDSTL
A          WOSLLST          16A          TEXT('List ID')
A          WOSLRC           1A          TEXT('Record code')
A          WOSLEID          16A          TEXT('Entry ID')
A          WOSLDESC         50A          TEXT('Description')
A          WOSLCS           8A          TEXT('System the +
                                     outfile was +
                                     created on')
A          WOSLCC           1A          TEXT('Century the +
                                     outfile was +
                                     created')
A          WOSLCD           6A          TEXT('Date the +
                                     outfile was +
                                     created')
A          WOSLCT           6A          TEXT('Time the +
                                     outfile was +
                                     created')
A          WOSLOW           10A         TEXT('Owner user +
                                     profile')
A*
A* End of specifications

```

Figure C-6. QAOSDSTO in Library QSYS Example

The meaning of each field in the record is:

- **WOSLLST** (List ID). The list ID of a distribution list contained in the directory. This is a two-part field, each part has 1 to 8 characters. This field is put into code from code page 500, character set 930.
- **WOSLRC** (Record code). Defines the type of record:
 - 0 - list header record
 - 1 - list entry record
- **WOSLEID** (Entry ID). This is the user ID and address of an entry contained in the distribution list when the record code is 1 (detail record). For record code 0 (header record), this field contains the local system name and system group on which this distribution list exists. This field is put into code from code page 500, character set 930.
- **WOSLDESC** (Entry description). This field contains the description associated with the entry ID. When the record code is 0 (header record), the entry description contains the list description. When the record code is 1 (detail record), the entry description contains the user description of the entry in the

list. If the user ID and address is in fact a remote distribution list, the entry description contains the remote list description.

- *WOSLCS* (System name). The system on which the output file was created.
- *WOSLCC* (Century). The century in which the output file was created (0=1900, 1=2000).
- *WOSLCD* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *WOSLCT* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *WOSLOW* (Owner User Profile). The profile of the user that owns the list.

Display Document Conversion (DSPDOCCVN) Output File Considerations

The Display Document Conversion (DSPDOCCVN) command can be used to create a database file containing information about document conversion programs that have been defined to OfficeVision/400.

The model output file supplied with the system is named QAO1DSPCVN in library QOFC.

The following data description specifications were used to create the model output file:

```
A***   DDS specifications for QOFC/QAO1DSPCVN   *****
A*                                           *
A           R DSPCVN                        TEXT('Record for displaying +
A                                           document conversions +
A                                           (DSPDOCCVN)')
A           DDCITP          10A             TEXT('Input document type')
A           DDCINM          5S 0           TEXT('Input document type +
A                                           number')
A           DDCRTP          10A             TEXT('Result document type')
A           DDCRNM          5S 0           TEXT('Result document type +
A                                           number')
A           DDCPGM          10A             TEXT('Conversion program +
A                                           name')
A           DDCLIB          10A             TEXT('Conversion program +
A                                           library')
A           DDDPCF          4A              TEXT('Handle PCFILE *YES/*NO')
A           DDCDSC          50A             TEXT('Document conversion +
A                                           description')
A           DDCSID          5S 0           TEXT('Coded character set ID')
A***   End of specifications   *****
```

The meaning of each field in the record is:

- *DDCITP* (Input document type). The name of the document type that this conversion accepts as input.
- *DDCINM* (Input document type number). The number associated with the input document type for this document conversion.
- *DDCRTP* (Result document type). The name of the document type that this conversion produces as output.

- *DDCRNM* (Result document type number). The number associated with the result document type for this document conversion.
- *DDCPGM* (Conversion program name). The name of the program to be called for this document conversion.
- *DDCLIB* (Conversion program library). The name of the library in which the document conversion program is located.
- *DDCPCF* (Handle PCFILE flag). A flag to specify whether this document conversion program accepts documents of type PCFILE in addition to the type specified in DDCITP (*YES=PCFILE is accepted, *NO=PCFILE is not accepted).
- *DDCDSC* (Document conversion description). The text description associated with the document conversion.
- *DDCSID* (Coded character set ID). The coded character set of the field DDCDSC (document conversion description).

Display Folder (DSPFLR) Output File Considerations

The Display Folder (DSPFLR) command can be used to create a database file that contains information about the documents within a folder or about the folders that are within a folder.

Folder Object Information

The model output file supplied with the system is named QADSPFLR in library QSYS and is used when DSPFLR TYPE(*FLR) OUTPUT(*OUTFILE) is specified.

The following data description specifications were used to create the model output file:

DSPFLR TYPE(*FLR) OUTPUT(*OUTFILE) Example

```
A* Start of specifications
A*
A* File Name: QADSPFLR in library QSYS
A* File Type: Physical
A* Record Format Name: FLRDTL
A*
A*
A          R FLRDTL
A          BASFLR          63A          TEXT('Folder path')
A          FLRNAM          12A          TEXT('Folder name')
A          DESC            44A          TEXT('Description')
A          LASTREV         8A          TEXT('Last revised')
A          CRTCEN          1A          TEXT('Century the +
                                     outfile was +
                                     created')
A          CRTDAT          6A          TEXT('Date the +
                                     outfile +
                                     was created')
A          CRTTIM          6A          TEXT('Time the +
                                     outfile was +
                                     created')
A          SYSNAM          8A          TEXT('System the +
                                     outfile was +
                                     created on')
A*
A* End of specifications
```

The meaning of each field in the record is:

- *BASFLR* (Folder path name). The folder path name used to locate the folder in the *FLRNAM* field.
- *FLRNAM* (Folder name). The name of each folder within the folder that is defined by the folder path name.
- *DESC* (Folder description). The description associated with each folder. This can contain blanks if no description was provided when the folder was created.
- *LASTREV* (Folder last revision date). This is the date the folder was last changed by either changing the folder description or by adding an object to or deleting an object from the folder. It is in the format MM/DD/YY, where MM=month, DD=day, and YY=year.
- *CRTCEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *CRTDAT* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *CRTTIM* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *SYSNAM* (System name). The system on which the output file was created.

Document Object Information

The other model output file supplied with the system is named QADSPDOC in library QSYS and is used when DSPFLR TYPE(*DOC) OUTPUT(*OUTFILE) is specified.

The following data description specifications were used to create the model output file:

```
A* Start of specifications
A*
A* File Name: QADSPDOC in library QSYS
A* File Type: Physical
A* Record Format Name: DOCCTL
A*
A          R DOCCTL
A          FLRNAM          63A      TEXT('Folder')
A          DOCNAM          12A      TEXT('Document')
A          DESC            44A      TEXT('Description')
A          LASTRV          8A       TEXT('Last revised')
A          DOCTYP          10A      TEXT('Document type')
A          CRTCEN          1A       TEXT('Century the +
                                     outfile was +
                                     created')
A          CRTDAT          6A       TEXT('Date the +
                                     outfile was +
                                     created')
A          CRTTIM          6A       TEXT('Time the +
                                     outfile was +
                                     created')
A          SYSNAM          8A       TEXT('System the +
                                     outfile was +
                                     created on')
A*
A* End of specifications
```

Figure C-7. DSPFLR TYPE(*DOC) OUTPUT(*OUTFILE) Example

The meaning of each field in the record is:

- *FLRNAM* (Folder path name). The folder path name used to locate the documents in the *DOCNAM* field.
- *DOCNAM* (Document name). The 1- to 12-character user assigned name of the document within the folder.
- *DESC* (Document description). The description associated with the document.
- *LASTRV* (Document last revision date). This is the date the document was either edited or details about the document (such as author, subject, or keywords) were changed. It is in the format MMDDYY, where MM=month, DD=day, and YY=year.
- *DOCTYP* (Document type). This is a keyword that describes the data stream content of the document. It can have the following values:
 - OTHER - unknown
 - FFTDCA - final-form text document
 - RFTDCA - revisable-form text document
 - PCFILE - personal computer data file
 - RFTS38 - IBM System/38 revisable document
 - HARDCOPY - reference to a printed document
 - RFT5520 - IBM 5520 revisable-form text document
 - EBCDIC - word processing EBCDIC text
 - WRDPRCINFF - word processing information file
 - IMAGE - image data subset document
 - TEXT3730 - IBM 3730 text data stream
 - DOCDESCDOC - document library descriptions document
 - DATAST3732 - IBM 3732 display document data stream
 - DOCUNITCTL - DIA defined document unit control
 - PRT1403 - IBM 1403 printer compatible data stream
 - AUDIO - digitized ADS audio (voice)
 - RFTAS400 - AS/400 revisable-form document
 - FFTAS400 - AS/400 final-form text
 - RFTDW4 - IBM DW4 revisable-form document
 - ASCIIIDATA - ASCII data
 - MODCA - mixed object document
 - 1 to 65535 - user-defined document type
- *CRTCEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *CRTDAT* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *CRTTIM* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *SYSNAM* (System name). The system on which the output file was created.

File Document (FILDOC) Output File Considerations

The File Document (FILDOC) command accepts a database file as input. The file can be a source file or data file with no office considerations. In this case, the data is read and filed into the document library.

The file could also be a data file that is formatted to be the same as the output file that the Retrieve Document (RTVDOC) command creates or the file can be a data file that is formatted to be the same as the output file that the Receive Distribution (RCVDST) command creates. When the file is in this form, the File Document command will recognize that and properly process the file to format the data and the interchange document profile (IDP).

Thus it is possible to issue a RTVDOC command to copy a document to a file or receive a distribution and later file that document back in the document library using the FILDOC command. It is also possible to write your own application to format a data file in the same manner as the Retrieve Document command or the Receive Distribution command does and use that file as input to the File Document command.

Fill Form Document (FILLFORM) Output File Considerations

When writing the data in the form fields to a database file, the system outfile shipped in the library QOFC and named QAOEFDTA is used as a model to create the output file specified in the OUTFILE parameter of the FILLFORM command. Unexpected results will occur if a record format other than the IBM-supplied format of the model file is used to create the file.

The following data description specifications were used to create the model output file:

```
A* Start of specifications
A*
A* File Name: QAOEFDTA in library QSYS
A* File Type: Physical
A* Record Format Name: QOEDATA
A*
A*
A      R QOEDATA                TEXT('Form field output +
A                                data file')
A      FDRCID          2A        TEXT('Record identifier =
A                                FD')
```

Figure C-8 (Part 1 of 2). Model Output File QAOEFDTA in library QSYS

| | | | | |
|--|----|-----------------------|-----|---------------------------|
| | A | FDSYSN | 8A | TEXT('System name') |
| | A | FDUSER | 10A | TEXT('User ID') |
| | A | FDJOB# | 10A | TEXT('Job name') |
| | A | FDJOB# | 6A | TEXT('Job number') |
| | A | FDTMSP | 13A | TEXT('Date/Time stamp') |
| | A | FDORSN | 1A | TEXT('Output reason') |
| | A | FDDOC | 12A | TEXT('Shell document') |
| | A | FDDFFID | 5A | TEXT('Form ID') |
| | A | FDDLGH | 3P | TEXT('Data length') |
| | A | FDNUM | 1A | TEXT('Numeric only') |
| | A | FDDCPS | 1P | TEXT('Decimal positions') |
| | A | FDNEGV | 1A | TEXT('Negative Value') |
| | A | FDDATA | 78A | TEXT('Form field data') |
| | A* | | | |
| | A* | End of specifications | | |

Figure C-8 (Part 2 of 2). Model Output File QAOEFDTA in library QSYS

The meaning of each field in the record is:

- **FDRCID** (Record ID). The record identifier constant. It is always be FD.
- **FDSYSN** (Document name). The name of the system that this record was created on.
- **FDUSER** (User ID). The user ID that was being used when the record was written to the file.
- **FDJOB#** (Job name). The name of the job that wrote this record.
- **FDJOB#** (Job number). The number of the job that wrote this record.
- **FDTMSP** (Transaction timestamp). The data and time this record was written. The format is CYYMMDDHHNNSS, where C = century, YY = year, MM = month, DD = day, HH = hour, NN = minute, and SS = seconds.
- **FDORSN** (Output reason code). The output reason specifies when the record was written to the file. Possible values are E = on exit, C = on cancel, and R = on refresh.
- **FDDOC** (Form document name). The name of the shell document.
- **FDDFFID** (Form field identifier). The form field ID specified in the form field instruction.
- **FDDLGH** (Form field length). The length of the data in the form field.
- **FDNUM** (Numeric only). The numeric data indicator, where Y = the data is numeric only and N = the data is character data.
- **FDDCPS** (Decimal positions). The number of decimal positions.
- **FDNEGV** (Negative value). The negative value indicator. A blank or zero (0) indicates the data is positive. A one (1) indicates the data is negative.
- **FDDATA** (Form field data). The form field data. The form field data is left justified in this field. If the form field is defined as numeric data only, the data is right justified within the defined length of the form field and filled with leading zeros. For example, if the form field is defined with a length of ten (10) and 123456 is entered, the data will be written as 0000123456 with the remaining sixty-eight positions in the field left blank.

Print Document (PRTDOC) Output File Considerations

The Print Document (PRTDOC) command can be used to create a database file that contains descriptive information about the document (such things as authors, subject, or action due date).

The model output file supplied with the system is named QAOPUFL in library QSYS.

The following data description specifications were used to create the model output file:

```
A* Start of specifications
A*
A* File Name: QAOPUFL in library QSYS
A* File Type: Physical
A* Record Format Name: RECORD
A*
A*
A      R RECORD
A      FIELD          256A
A*
A* End of specifications
```

The meaning of each field in the record is:

- *FIELD* (Data field). See the following tables for how the data field is defined for each record written to the file.

File Format for PRTFILE Files

For each document copied to an output file, one header record is created as the first record in that file. All records in the output file have a record length of 256 bytes.

Header Record

The header record contains the following fields:

Figure C-9. PRTFILE Header Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|------------------|--------------|-----------------------------|---|
| 1 | 1 | B | X'20' |
| 2 | 2 | C | PF |
| 4 | 8 | C | Original system name |
| 12 | 1 | C | Century created |
| 13 | 6 | C | Date created |
| 19 | 6 | C | Time created |
| 25 | 232 | C | Unused (unused positions are set to blanks) |

Following the header record are the document records (record identifiers 10 through 99). There will be a start document record as the first record for each document printed to the file.

Start Document Record

The start document record contains the following fields:

Figure C-10. PRTFILE Start Document Record

| 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 |
| 27 | 62 | C | Subdirectory |
| 89 | 12 | C | 12-character folder name |
| 101 | 156 | C | Unused (unused positions are set to blanks) |

Document Description Records

Following the document record are document description records. The first document description record contains the following fields:

Figure C-11. First PRTFILE Document Description Record

| 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 | Unused (unused positions are set to blanks) |
| 134 | 16 | C | Document class |
| 150 | 8 | C | Date complete |
| 158 | 8 | C | Date last revised |
| 166 | 8 | C | Creation date |
| 174 | 8 | C | Action due date |
| 182 | 8 | C | Expiration date |
| 190 | 10 | C | Project |
| 200 | 10 | C | Reference |
| 210 | 16 | C | Keyword |
| 226 | 1 | C | Status |
| 227 | 2 | B | Internal status ('8000'X=print as labels, '4000'X=object (graphics), '0000'X=other) |
| 229 | 6 | C | Time last revised |
| 235 | 22 | C | Unused (unused positions are set to blanks) |

The second document description record contains the following fields:

Figure C-12 (Page 1 of 2). Second PRTFILE Document Description Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|------------------|--------------|-----------------------------|---|
| 1 | 2 | C | Record identifier (12) |
| 3 | 35 | C | Document description |
| 38 | 60 | C | Capitalized subject |
| 98 | 13 | C | Original system |
| 111 | 44 | C | Original name |
| 155 | 4 | C | Profile code page |
| 159 | 4 | C | Document code page |
| 163 | 1 | C | Unused (unused positions are set to blanks) |
| 164 | 1 | C | Unused (unused positions are set to blanks) |
| 165 | 2 | B | Document type |
| 167 | 17 | C | Unused (unused positions are set to blanks) |

Figure C-12 (Page 2 of 2). Second PRTFILE Document Description Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|------------------|--------------|-----------------------------|--|
| 184 | 8 | C | Unused (unused positions are set to blanks) |
| 192 | 8 | C | Unused (unused positions are set to blanks) |
| 200 | 6 | C | Unused (unused positions are set to blanks) |
| 206 | 1 | C | Unused (unused positions are set to blanks) |
| 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 |
| 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 (unused positions are set to blanks) |
| 212 | 1 | C | Unused (unused positions are set to blanks) |
| 213 | 44 | C | Unused (unused positions are set to blanks) |

The third document description record contains the following fields:

Figure C-13. Third PRTFILE Document Description Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|------------------|--------------|-----------------------------|-------------------------|
| 1 | 2 | C | Record identifier (13) |
| 3 | 16 | C | Document owner |
| 19 | 8 | C | Date filed |
| 27 | 4 | C | Time filed |
| 31 | 8 | C | User document date |
| 39 | 60 | C | Reference |
| 99 | 20 | C | Status |
| 119 | 60 | C | Cabinet reference |
| 179 | 1 | C | Archive mark |
| 180 | 1 | C | Work in progress |
| 181 | 72 | C | Unused (set to blanks) |

The fourth document description record contains the following fields:

Figure C-14 (Page 1 of 2). Fourth PRTFILE Document Description Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|------------------|--------------|-----------------------------|--|
| 1 | 2 | C | Record identifier (14) |
| 3 | 240 | C | 60-character subject can occur 4 times |

Figure C-14 (Page 2 of 2). Fourth PRTFILE Document Description Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|-------------------------|---------------------|------------------------------------|--------------------------------|
| 243 | 14 | C | Unused (set to blanks) |

The fifth document description record contains the following fields:

Figure C-15. Fifth PRTFILE Document Description Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|-------------------------|---------------------|------------------------------------|--|
| 1 | 2 | C | Record identifier (15) |
| 3 | 240 | C | 20-character author can occur 12 times |
| 243 | 14 | C | Unused (set to blanks) |

The sixth document description record contains the following fields:

Figure C-16. Sixth PRTFILE Document Description Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|-------------------------|---------------------|------------------------------------|---|
| 1 | 2 | C | Record identifier (16) |
| 3 | 245 | C | 35-character keywords can occur 7 times |
| 248 | 9 | C | Unused (set to blanks) |

The seventh document description record contains the following fields:

Figure C-17. Seventh PRTFILE Document Description Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|-------------------------|---------------------|------------------------------------|--|
| 1 | 2 | C | Record identifier (17) |
| 3 | 240 | C | 60-character recipient can occur 4 times |
| 243 | 14 | C | Unused (set to blanks) |

General Format Record

The general format record contains the following fields:

Figure C-18. PRTFILE General Format Record

| 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 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 |
| 198 | 8 | C | Query library |
| 206 | 2 | B | Graphic Character Set ID |
| 208 | 2 | B | Code Page Global ID |
| 210 | 10 | C | Full 10 character field/query name |
| 220 | 10 | C | Full 10 character query library |
| 230 | 27 | C | Unused (unused positions are set to blanks) |

Page Format Record

The page format record contains the following fields:

Figure C-19. PRTFILE Page Format Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|------------------|--------------|-----------------------------|---|
| 1 | 2 | C | Record identifier (30) |
| 3 | 2 | B | Format ID number |
| 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 | 4 | B | Unused |
| 101 | 3 | C | Degrees to rotate, blanks imply automatic |
| 104 | 153 | C | Unused (unused positions are set to blanks) |

Line Format Record

The line format record contains the following fields:

Figure C-20 (Page 1 of 2). PRTFILE Line Format Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|------------------|--------------|-----------------------------|--|
| 1 | 2 | C | Record identifier (40) |
| 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 | Automatic 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 types (1 character 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 |

Figure C-20 (Page 2 of 2). PRTFILE Line Format Record

| 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 through 79). If more than 235 bytes of text are 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).

If a line contains only a carrier return or a required carrier return, a record is not created. The line number in the next record is incremented to account for the blank lines. Exceptions include the following:

- Lines with line numbers
- Lines with change flags
- Lines with left margins greater than the document farthest left margin

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, and so on) |
| 9 | 1 | C | Margin text (1 = yes, 2 = no) |
| 10 | 2 | B | Position of last non-blank character in line |
| 12 | 1 | C | Required carrier return (1 = yes, 2 = no) |

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|------------------|--------------|-----------------------------|---|
| 13 | 9 | C | Unused (unused positions are set to blanks) |
| 22 | 235 | C | Line |

Text Bold Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|--|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (51) ¹ |
| 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

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|--|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (52) ¹ |
| 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) Overstrike Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|--|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (53) ¹ |
| 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) Backspace Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|--|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (54) ¹ |
| 1 The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

Text Base Line Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|---|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (60) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

Text Base Bold Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|---|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (61) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

Text Base Underline Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|---|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (62) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

Text Base Overstrike Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|---|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (63) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

Text Base Backspace Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|---|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (64) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

Text Half Index Down (Subscript) Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|---|--------------|-----------------------------|-------------------------------------|
| 1 | 2 | C | Record identifier (70) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

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) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

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) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

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) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

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) ¹ |
| ¹ The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record." | | | |

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 following fields:

Figure C-21. PRTFILE Instruction Record

| 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 (can contain values less than '40'X) |

Start Error Log Record

The start error log record contains the following fields:

Figure C-22. PRTFILE Start Error Log Record

| 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 | 12-character folder name |
| 101 | 156 | C | Unused (unused positions are set to blanks) |

End Document Record

The end document record contains the following fields:

Figure C-23 (Page 1 of 2). PRTFILE End Document Record

| 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 |

Figure C-23 (Page 2 of 2). PRTFILE End Document Record

| Beginning Column | Field Length | Character (C) or Binary (B) | Contents or Description |
|-------------------------|---------------------|------------------------------------|---|
| 89 | 12 | C | 12-character folder name |
| 101 | 156 | C | Unused (unused positions are set to blanks) |

Query Calendar Item (QRYCALITM) Output File Considerations

The Query Calendar Item (QRYCALITM) command can be used to produce a database file that contains information about all items that match the specified query criteria. The key fields included are the calendar name (QCCAL), calendar owner names (QCODEN and QCODGN), and item identifier (QCIID). These fields uniquely identify the item for use with other commands (such as DSPCALITM).

The text line fields (QCTXT1, QCTXT2, and QCTXT3) are set to the first three text lines that are associated with the calendar item. For meetings, the first text line is the subject, the second line is the meeting place, and the third line is the first line of the meeting purpose. If the item is for a deleted meeting (QCSTS = *DELETED), then the text lines are for the meeting entry instead of the meeting. When there are more text lines (QCNTXT greater than 3), use the DSPCALITM command to obtain the additional lines.

The following fields are blank (or zero for numeric fields) if the record is added by a user who has only *TIMES authority to the item: QCISEC, QCRMSG, QCRDAT, QCRTIM, QCSTS, QCRDEN, QCRDGN, QCJOB, QCJLIB, QCJCMD, QCPPTY, QCNTXT, QCTXT1, QCTX1C, QCTX1P, QCTXT2, QCTX2C, QCTX2P, QCTXT3, QCTX3C, and QCTX3P.

The model output file supplied with the system is named QAOYQRYITM and is located in library QOFC. Use the following data description specifications to create the model output file:

```

A*** DDS specifications for QOFC/QAOYQRYITM *****
A*
A          R QRYCAL                TEXT('Record for querying +
A                                calendar items (QRYCALITM)')
A          QCCEN                    1A    TEXT('Century')
A          QCDAT                    6A    TEXT('Date (YYMMDD)')
A          QCTIM                    6A    TEXT('Time (HHMMSS)')
A          QCCALC                   4S 0  TEXT('Calendar name character +
A                                set')
A          QCCALP                   4S 0  TEXT('Calendar name code page')
A          QCCAL                   10A    TEXT('Calendar name')
A          QCODEN                   8A    TEXT('Calendar owner user ID')
A          QCODGN                   8A    TEXT('Calendar owner address')
A          QCIID                   20A    TEXT('Item identifier')
A          QCITYP                   9A    TEXT('Item type')
A          QCISEC                   13A   TEXT('Item security')
A          QCRMSG                   4A    TEXT('Reminder message or job +
A                                or S/36 procedure pending')
A          QCRDAT                   6A    TEXT('Reminder message date +
A                                - job format') +
A          QCRTIM                   5A    TEXT('Reminder msg time +
A                                (HH:MM)')
A          QCDATE                   6A    TEXT('Item date (YYMMDD)')
A          QCSTIM                   5A    TEXT('Item start time (HH:MM)')
A          QCETIM                   5A    TEXT('Item end time (HH:MM)')
A          QCSTS                   10A   TEXT('Meeting/event status')
A          QCRDEN                   8A    TEXT('Meeting requester +
A                                user ID')
A          QCRDGN                   8A    TEXT('Meeting requester +
A                                address')

```

Figure C-24 (Part 1 of 2). Model Output File for QAOYQRYITM in library QOFC

```

A          QCJOB          10A          TEXT('Scheduled job description +
A          QCJLIB          10A          TEXT('Job description library +
A          QCJCMD          256A         TEXT('Scheduled job command +
A          QCPPTY          1A          TEXT('S/36 procedure priority')
A          QCNTXT          3S 0        TEXT('Number of text lines')
A          QCTXT1          50A         TEXT('Text line 1')
A          QCTX1C          4S 0        TEXT('Text line 1 +
A          QCTX1P          4S 0        TEXT('Text line 1 +
A          QCTXT2          50A         TEXT('Text line 2')
A          QCTX2C          4S 0        TEXT('Text line 2 +
A          QCTX2P          4S 0        TEXT('Text line 2 +
A          QCTXT3          50A         TEXT('Text line 3')
A          QCTX3C          4S 0        TEXT('Text line 3 +
A          QCTX3P          4S 0        TEXT('Text line 3 +
A          QCBUSY          4A          TEXT('Busy item +
A          (*YES/*NO)')
A*** End of specifications *****

```

Figure C-24 (Part 2 of 2). Model Output File for QAOYQRYITM in library QOFC

Note: On double-byte systems the fields QCCAL, QCJCMD, QCTXT1, QCTXT2, and QCTXT3 have a data type of O (open). These fields can contain mixed single-byte and double-byte data on these systems.

The meaning of each field in the record is:

- *QCCEN* (Century). The century in which the output file was created (0=1900, 1=2000).
- *QCDAT* (Job date). The date the output file was created. Specify the date as YYMMDD, where YY=year, MM=month, and DD=day.
- *QCTIM* (Job time). The time the output file was created. Specify the time as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *QCCALC* (Calendar name character set). The calendar name field (QCCAL), which is stored in the output file in this character set.
- *QCCALP* (Calendar name code page). The calendar name field (QCCAL), which is stored in the output file in this code page.
- *QCCAL* (Calendar name). The 1- to 10-character name assigned to the calendar when the user created it.
- *QCODEN* (Calendar owner user ID). The user ID of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *QCODGN* (Calendar owner address). The address of the calendar's owner. This is stored in the output file in character set 930, code page 500.
- *QCIID* (Item identifier). The item identifier, a 20-character string which, when combined with the calendar name and owner name can uniquely identify an

item. Use the item identifier as input to other commands (such as DSPCALITM) to process this specific item.

- *QCITYP* (Item type). The type of calendar item. Possible values are *REMINDER, *EVENT, *MEETING, *JOB, or *S36PRC.
- *QCISEC* (Item security). The item security. Possible values are *UNCLASSIFIED, *CONFIDENTIAL, or *PERSONAL. If the item is a meeting, then this security value is for the meeting, not the meeting entry. If the meeting has been deleted (QCSTS = *DELETED), the security value is for the meeting entry.
- *QCRMSG* (Reminder message or job pending flag). If the item is an event, reminder, or meeting, this flag indicates whether a reminder message is sent for the item. It is set to *YES if a reminder message is sent and *NO if a reminder message is not sent (or has already been sent). If the item is a job or System/36 procedure, this flag indicates whether the job or procedure is still pending (has not been submitted). The flag is set to *YES if the job or procedure is has not yet been submitted and *NO if the job or procedure has already been submitted.
- *QCRDAT* (Reminder message date). The date the reminder message for this item is scheduled to be sent. Specify the date as YYMMDD, where YY=year, MM=month, and DD=day. This field is blank when the item is a job or System/36 procedure, or when the reminder message flag field (QCRMSG) is set to *NO.
- *QCRTIM* (Reminder message time). The time the reminder message for this item is scheduled to be sent. Specify the time as HH:MM, where HH=hour and MM=minutes. This field is blank when the item is a job or System/36 procedure, or when the reminder message flag field (QCRMSG) is set to *NO.
- *QCDATE* (Item date). The date the calendar item is scheduled on the calendar. Specify the date as YYMMDD, where YY=year, MM=month, and DD=day.
- *QCSTIM* (Item start time). The time at which the calendar item is scheduled to start. Specify the time as HH:MM, where HH=hour and MM=minutes. This field is blank if the item is a reminder, because reminders do not have a start time.
- *QCETIM* (Item end time). The time at which the calendar item is scheduled to end. Specify the time as HH:MM, where HH=hour and MM=minutes. This field is blank if the item is a reminder, job, or System/36 procedure, because these items do not have an end time.
- *QCSTS* (Item status). The meeting or event item status. If the item is an event, this field is either *TENTATIVE or *CONFIRMED. If the item is a meeting, this field is *TENTATIVE, *CONFIRMED, *POSTPONED, *CANCELED, or *DELETED. This field is blank for reminders, jobs, and System/36 procedures.
- *QCRDEN* (Meeting requester user ID). The user ID of the meeting requester. This is stored in the output file in character set 930, code page 500. If the item is not a meeting this field is blank.
- *QCRDGN* (Meeting requester address). The address of the meeting requester. This is stored in the output file in character set 930, code page 500. If the item is not a meeting, this field is blank.

- *QCJOB* (Job description or System/36 procedure name). If the item is a job, this field contains the name of the job description to be used when the job is submitted. If the item is a System/36 procedure, this field contains the name of the procedure to be submitted. This field is blank for events, reminders, and meetings.
- *QCJLIB* (Job description library or System/36 procedure library). If the item is a job, this field contains the library where the job description (field *QCJOB*) is located. If the item is a System/36 procedure, this field contains the library where the procedure (field *QCJOB*) is located. This field is blank for events, reminders, and meetings.
- *QCJCMD* (Job command or System/36 procedure parameters). If the item is a job, this field contains the job command string to be submitted. If the item is a System/36 procedure, this field contains the parameter string to be submitted with the procedure. This field is blank for events, reminders, and meetings.
- *QCPPTY* (System/36 procedure priority). If the item is a System/36 procedure, this field contains the priority that the procedure should be submitted with. This field is blank for events, reminders, meetings, and jobs.
- *QCNTXT* (Number of text lines). The number of lines of text for the item. For meetings, this includes the meeting text and any meeting entry text. A meeting that has not been deleted has at least 3 lines of text (subject, place, and purpose), even if they are blank, while a deleted meeting has no lines of text. An event, reminder, meeting entry, job, or System/36 procedure has at least 1 line of text even if the line is blank.
- *QCTXT1* (Text line 1). The first line of text for the item.
- *QCTX1C* (Text line 1 character set). The first line of text field *QCTXT1*, which is stored in the output file in this character set.
- *QCTX1P* (Text line 1 code page). The first line of text field *QCTXT1*, which is stored in the output file in this code page.
- *QCTXT2* (Text line 2). The second line of text for the item.
- *QCTX2C* (Text line 2 character set). The second line of text field *QCTXT2*, which is stored in the output file in this character set.
- *QCTX2P* (Text line 2 code page). The second line of text field *QCTXT2*, which is stored in the output file in this code page.
- *QCTXT3* (Text line 3). The third line of text for the item.
- *QCTX3C* (Text line 3 character set). The third line of text field *QCTXT3*, which is stored in the output file in this character set.
- *QCTX3P* (Text line 3 code page). The third line of text field *QCTXT3*, which is stored in the output file in this code page.
- *QCBUSY* (Busy item flag). The flag that indicates whether the item represents a time on the calendar that should be considered busy. The flag is set to *YES if the item represents a time on the calendar that should be considered busy; otherwise it is set to *NO. A reminder, job, or System/36 procedure never represents busy times on a calendar, while an event represents a busy time. A meeting represents a busy time if the status field (*QCSTS*) is *TENTATIVE or *CONFIRMED and the invitee's status is *ATTENDING, *NOTATTENDING, or *NOTINVITED.

Query Document Library (QRYDOCLIB) Output File Considerations

The Query Document Library (QRYDOCLIB) command can be used to create a database file that contains information about documents in the document library that have satisfied some query criteria.

The model output file supplied with the system is named QAOSIQDL in library QSYS.

The following data description specifications were used to create the model output file:

```
A* Start of specifications
A*
A* File Name: QAOSIQDL in library QSYS
A* File Type: Physical
A* Record Format Name: OSQDL
A*
A*
A      R OSQDL
A      QDLSNO      6S      TEXT('Sequence number')
A      QDLDID      24A     TEXT('Document +
A                               identifier')
A      QDLDNM      12A     TEXT('Document name')
A      QDLFLR      63A     TEXT('Folder name')
A      QDLONM      10A     TEXT('Object name')
A      QDLUID      8A      TEXT('Owner user ID')
A      QDLADR      8A      TEXT('Owner user +
A                               address')
A      QDLDSZ      8S      TEXT('Document size')
A      QDLDTP      5S      TEXT('Document type')
A      QDLSYC      13A     TEXT('System code')
A      QDLACF      1A      TEXT('Access flag')
A      QDLPER      1A      TEXT('Personal flag')
A      QDLWIP      1A      TEXT('Work in process flag')
A      QDLAHV      1A      TEXT('Mark for offline flag')
A      QDLCKO      1A      TEXT('Checkout flag')
A      QDLCRS      3S      TEXT('Data character set')
A      QDLCPG      3S      TEXT('Data code page')
A      QDLDL       3S      TEXT('Data length')
A      QDLRC       3S      TEXT('Record code')
A      QDLDTA      500A    TEXT('Data')
A      QDLOSZ      8S      TEXT('Document Object Size')
A      QDLC SY      8A      TEXT('System the outfile +
A                               was created on')
A      QDLCDC      1A      TEXT('Century the outfile +
A                               was created')
A      QDLCDD      6A      TEXT('Date the outfile +
A                               was created')
A      QDLCDM      6A      TEXT('Time the outfile +
A                               was created')
A      QDLCSX      5S      TEXT('Data character set')
A      QDLCPX      5S      TEXT('Data code page')
A*
A* End of specifications
```

Figure C-25. QAOSIQDL in Library QSYS Example

The meaning of each field in the record is:

- *QDLSNO* (Sequence number). A zoned numeric field that starts at 1 for the first record of document descriptor data and is incremented by 1 for the next record of a description.
- *QDLDDID* (Document identifier). This is the name assigned to the document by the system when it was filed. It consists of the system name concatenated with a time stamp. No two documents have the same document identifier. Each record for a document has the same document identifier. The Document Interchange Architecture refers to the document identifier as library-assigned document name (LADN). It is in the format
YYYYMMDDHHMNSSHSSNSNSNSNSNSNSNSNSN where:

YYYY = year
MM = month
DD = day
HH = hour
MN = minute
SS = second
HS = hundredths of second
SNSNSNSNSNSNSNSNSN = system name

- *QDLDNM* (Document name). This is the unique document name specified by the user when the document was filed into the document library. If the document is not in a folder, this field contains the object name.
- *QDLFLR* (Folder name). This is the folder path name to the folder containing the document. If the document is not in a folder, this field contains blanks.
- *QDLONM* (Object name). This is the unique object name created by the system and used on other system commands (for example, DSPOBJD, DMPSYSOBJ).
- *QDLUID* (Owner user ID). The user ID of the owner of the document. This data is put into code from code page 256, character set 930.
- *QDLADR* (Owner address). The address of the owner of the document. This data is put into code from code page 256, character set 930.
- *QDLDSZ* (Document size). This is the size in bytes of the document data unless the document is FFTAS400 or RFTAS400. If the document is FFTAS400 or RFTAS400, this field will be zero. If a conversion has been performed on the document, the size of the converted data will then be given. To get the size of the IDP in the document, take the sum of all the QDLDL fields when the field QDLRC is equal to record code 500. To get the object size of the document object refer to the QDLOSZ (Object Size) field.

- **QDL DTP** (Document type). The document data is defined by this identifier.

| DOCUMENT TYPE VALUE | DOCUMENT TYPE MEANING |
|---------------------|--|
| 2 | Final-form text document (FFTDCA) |
| 3 | 5520 revisable-form text document (RFT5520) |
| 4 | Word-processing EBCDIC (EBCDIC) |
| 5 | Word-processing information file (WRDPRCINFF) |
| 6 | Image-data subset document (IMAGE) |
| 7 | 3730 text data stream (TEXT3730) |
| 8 | DIA document library document descriptor document (DOCDESCDOC) |
| 9 | 3732 display document data stream (DATAST3732) |
| 10 | DIA defined document unit content (DOCUNITCTL) |
| 11 | Revisable-form text document (RFTDCA) |
| 12 | 1403 printer compatible data stream with variable length, unblocked records(PRT1403) |
| 13 | Digitized ADS audio (AUDIO) |
| 14 | IBM PC data file (PCFILE) |
| 15 | Hard copy of document (HARDCOPY) |
| 16 | Mixed object document (MODCA) |
| 18 | No document content |
| 46 | Enterprise meeting notice (EMN) |
| 223 | Data file |
| 32753 | List of documents created from search (DOCLIST) |
| 32768 | AS/400 revisable-form document (RFTAS400) |
| 32769 | AS/400 final-form text (FFTAS400) |
| 32770 | IBM DW4 revisable-form document (RFTDW4) |
| 32912 | ASCII data (ASCIIDATA) |
| 65262 | IBM S/38 revisable document (RFT38) |

Also, refer to the *Interchange Document Profile Reference* manual for more details.

- **QDL SYC** (Document system code). The document data is defined by this identifier. See the File Document (FILDOC) command for further information. This data is put into code from code page 256, character set 337.
- **QDL ACF** (Access flag). A 1-character flag to indicate if the document is a private or public document.
 - 0 =public (Access code 0000 assigned to document)
 - 1 =private (Access code 0000 not assigned to document)
- **QDL PER** (Personal flag). A 1-character flag to indicate if the document is personal (not available to users working on behalf of the document owner).
 - 0 =not personal
 - 1 =personal
- **QDL WIP** (Work in process flag). A 1-character flag to indicate a revisable (work in process document) or a final (non-replaceable) document.
 - 0 =final (Once set to final, cannot be changed to revisable)
 - 1 =revisable
- **QDL AHV** (Mark for offline flag). A 1-character flag to indicate if a document has been marked for saving to offline storage.
 - 0 =not marked for save
 - 1 =marked for save only
 - 2 =marked for save and delete document object

– 3 =marked for save and delete all data

- **QDLCKO** (Checkout flag). A 1-character flag to indicate if a revisable document has been checked out for update.
 - 0 =not checked out
 - 1 =checked out for update.
- **QDLCRS** (Data field character set). The data in the data field (**QDLDTA**) is put into code in this character set. This field is set to 999 if the character set is greater than 999. **QDLCSX** can be used to get the data field character set.
- **QDLCPG** (Data field code page). The data in the data field (**QDLDTA**) is put into code in this code page. This field is set to 999 if the code page is greater than 999. **QDLCPX** can be used to get the data field code page.
- **QDLDDL** (Length of data). This number defines the length of the data field (**QDLDTA**). It encompasses only the data field. For some records, the length is not really necessary because the following data is a fixed length. For variable length data it is necessary. It is correctly set for all records.
- **QDLRC** (Record code). A unique zoned number is defined for each record type. The record code must be used to determine the definition of the data to follow.
- **QDLDTA** (Data field). This field takes on many meanings and is further described in the following:

| Record Code | Data Field | Description |
|-------------|------------|--|
| 105 | QDLDTA | Document description. This is a 1- to 44-character document name assigned by the user. The system applies no meaning to this description. QDLCRS and QDLCPG define the character set and code page of this data. |
| 110 | QDLDTA | Creation date and time. This is the date and time the document was created. This date and time are contained in 12 characters in the format YYYYMMDDHHMN where: YYYY = year MM = month DD = day HH = hour MN = minute |
| 115 | QDLDTA | Expiration date. This is the date when the document is no longer needed. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |
| 120 | QDLDTA | Document date. This is any unique date the user wishes to associate with the document. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |

| Record Code | Data Field | Description |
|-------------|------------|--|
| 125 | QDLDTA | File date and time. This is the date and time the document was filed in the system's document library. This date and time were automatically created by the system when the document was filed. This date and time are contained in 16 characters in the format YYYYMMDDHHMNSSHS where: YYYY = year MM = month DD = day HH = hour MN = minute SS = second HS = hundredths of second |
| 130 | QDLDTA | Date and time last changed. This is the date and time the document content or document details were last changed. This date and time are contained in 16 characters in the format YYYYMMDDHHMNSSHS where: YYYY = year MM = month DD = day HH = hour MN = minute SS = second HS = hundredths of second |
| 135 | QDLDTA | Action due date. This is the date when the action specified in the document is due. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |
| 140 | QDLDTA | Completion date. This is the date when the action specified in the document is complete. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |
| 145 | QDLDTA | Author name. This is the name of the author of the document (1 to 20 characters). QDLCRS and QDLCPG define the character set and code page of this data. |
| 150 | QDLDTA | Copy name. This is the name of one entry in the copy list (1 to 60 characters). QDLCRS and QDLCPG define the character set and code page of this data. |
| 155 | QDLDTA | Document class. This is the document class (1 to 16 characters). QDLCRS and QDLCPG define the character set and code page of this data. |
| 160 | QDLDTA | File cabinet reference. This refers to offline storage where a printed document resides (1 to 60 characters). QDLCRS and QDLCPG define the character set and code page of this data. |
| 165 | QDLDTA | Subject. This is the subject of the document (1 to 60 characters). QDLCRS and QDLCPG define the character set and code page of this data. |

| Record Code | Data Field | Description |
|-------------|------------|---|
| 170 | QDLDTA | Keyword. This is a keyword associated with the document (1 to 35 characters). QDLCRS and QDLCPG define the character set and code page of this data. |
| 175 | QDLDTA | Reference. This is the reference associated with the document (1 to 60 characters). QDLCRS and QDLCPG define the character set and code page of this data. |
| 180 | QDLDTA | Status. This is the user-defined work progress descriptor associated with the document (1 to 20 characters). QDLCRS and QDLCPG define the character set and code page of this data. |
| 185 | QDLDTA | Project. This is the user-defined work identifier associated with the document (1 to 10 characters). QDLCRS and QDLCPG define the character set and code page of this data. |
| 190 | QDLDTA | Index date and time. This is the date and time the document was last indexed. This date and time are contained in 16 characters in the format YYYYMMDDHHMNSSHS where: YYYY = year MM = month DD = day HH = hour MN = minute SS = second HS = hundredths of second |
| 195 | QDLDTA | Revision date and time. This is the date and time the document content was last changed. This date and time are contained in 16 characters in the format YYYYMMDDHHMNSSHS where: YYYY = year MM = month DD = day HH = hour MN = minute SS = second HS = hundredths of second |
| 500 | QDLDTA | IDP data. This is the interchange document profile (IDP) that is kept with the document. It conforms to the data stream definition of the Document Interchange Architecture. Refer to the <i>Interchange Document Profile Reference</i> manual for more information. This data is optionally provided for the user that needs to access the IDP for fields other than those provided by separate record codes. It is the user's responsibility to parse this data and extract the desired information. This field will be 500 bytes of the IDP per record. The last record with this record code can have less than 500 bytes. The data length is defined by QDLDL. |

- **QDLOSZ** (Object Size). The document object size. This is the size in bytes of the amount of space the document is using in the system. It includes the IDP, data stream, byte string space, internal information, or text internal data.
- **QDLCSY** (System name). The system on which the output file was created.

- *QDLCDC* (Century). The century in which the output file was created (0=1900, 1=2000).
- *QDLCDD* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *QDLCDM* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *QDLCSX* (Data field character set). The data in the data field is in this character set.
- *QDLCPX* (Data field code page). The data in the data field is in this code page.

Query Distribution (QRYDST) Output File Considerations

The Query Distribution (QRYDST) command can be used to create a database file that contains information about incoming distributions currently queued by the system waiting for the recipient to obtain them. The Query Distribution command can also be used to create a database file that contains information about distributions the user has sent.

Incoming Distribution Information

The model output file supplied with the system is named QAOSILIN in library QSYS and is used for incoming distribution information.

The following data description specifications were used to create the model output file:

```

A* Start of specifications
A*
A* File Name: QAOSILIN in library QSYS
A* File Type: Physical
A* Record Format Name: OSLIN
A*
A*
A      R OSLIN
A      LINSNO      6S      TEXT('Sequence number')
A      LINDID      20A     TEXT('Distribution +
A                          identifier')
A      LINRUI      8A      TEXT('Recipient identifier')
A      LINRUA      8A      TEXT('Recipient address')
A      LINSUI      8A      TEXT('Sender identifier')
A      LINSUA      8A      TEXT('Sender address')
A      LINSDT      8A      TEXT('Send date')
A      LINSTM      6A      TEXT('Send time')
A      LINPER      1A      TEXT('Personal flag')
A      LINPTY      1A      TEXT('Priority flag')
A      LINCOD      1A      TEXT('Confirmation of +
A                          delivery flag')
A      LINDTP      5S      TEXT('Document type')
A      LINSYC      13A     TEXT('System code')
A      LINCID      3S      TEXT('Distribution +
A                          description character set')
A      LINCPCG     3S      TEXT('Distribution +
A                          description code page')
A      LINDSD      44A     TEXT('Distribution +
A                          description')
A      LINDEX      2S      TEXT('Distribution +
A                          identifier extension')
A      LINMLS      1A      TEXT('Mail log entry status')
A      LINCDD      8A      TEXT('Last date changed')
A      LINCTM      6A      TEXT('Last time changed')
A      LINCSDY     8A      TEXT('System the outfile +
A                          was created on')
A      LINCDC      1A      TEXT('Century the outfile +
A                          was created')
A      LINCDD      6A      TEXT('Date the outfile +
A                          was created')
A      LINCDDM     6A      TEXT('Time the outfile +
A                          was created')
A      LINTCS      5S      TEXT('Description +
A                          character set')
A      LINTCP      5S      TEXT('Description +
A                          code page')
A      LINTXD      44A     TEXT('Description')
A      LINCSDX     5S      TEXT('Distribution +
A                          document character set')
A      LINCPCX     5S      TEXT('Distribution +
A                          document code page')
A      LINNTE      1A      TEXT('Note flag')
A*
A* End of specifications

```

Figure C-26. QAOSILIN in Library QSYS Example

The meaning of each field in the record is:

- *LINSNO* (Sequence number). A zoned numeric field that starts at 1 for the first record of a distribution and is incremented by 1 for the next record of a distribution. Note that this number continues to be incremented. It does not start over when the records of another distribution are written to the file.
- *LINDID* (Distribution identifier). The unique identifier assigned to a distribution. It is composed of the user's address (8 characters), the user ID (8 characters), and a 4-byte zoned sequence number with leading zeros. For example: NEWYORK JACKSON 0023. Each record for a distribution has the same distribution ID. This data is put into code from code page 256, character set 930.
- *LINRUI* (Recipient user ID). This is the user ID of the person who receives the distribution. This data is put into code from code page 256, character set 930.
- *LINRUA* (Recipient user address). This is the address of the person who receives the distribution. This data is put into code from code page 256, character set 930.
- *LINSUI* (Sender user ID). This is the user ID of the person who sent the distribution. This data is put into code from code page 256, character set 930.
- *LINSUA* (Sender address). This is the address of the person who sent the distribution. This data is put into code from code page 256, character set 930.
- *LINSDT* (Date distribution sent). This is the date that the distribution was sent by the sender. If the sender is on another system in the office network, this is the date from that system. It is in the format YYYYMMDD where:
 - YYYY = year
 - MM = month
 - DD = day
- *LINSTM* (Time distribution sent). This is the time that the distribution was sent by the sender. If the sender is on another system in the office network, this is the time from that system. It is in the format HHMNSS where:
 - HH = hour
 - MN = minute
 - SS = second
- *LINPER* (Personal distribution). A value of 1 indicates this is a personal distribution; 0 indicates it is not personal.
- *LINPTY* (Distribution priority). A value of 1 indicates this was sent with high priority; 0 indicates it was sent with normal priority.
- *LINCOD* (Confirmation of delivery). A value of 1 indicates that the sender asked for confirmation of delivery; 0 indicates that confirmation of delivery was not requested.

- *LINDTP* (Document type). The document data is defined by this identifier.

| DOCUMENT TYPE VALUE | DOCUMENT TYPE MEANING |
|---------------------|---|
| 2 | Final-form text document (FFTDCA) |
| 3 | 5520 revisable-form text document (RFT5520) |
| 4 | Word-processing EBCDIC (EBCDIC) |
| 5 | Word-processing information file (WRDPRCINFF) |
| 6 | Image-data subset document (IMAGE) |
| 7 | 3730 text data stream (TEXT3730) |
| 8 | DIA document library document descriptor document (DOCDESCDOC) |
| 9 | 3732 display document data stream (DATAST3732) |
| 10 | DIA defined document unit content (DOCUNITCTL) |
| 11 | Revisable-form text document (RFTDCA) |
| 12 | 1403 printer compatible data stream with variable length, unblocked records (PRT1403) |
| 13 | Digitized ADS audio (AUDIO) |
| 14 | IBM PC data file (PCFILE) |
| 15 | Hard copy of document (HARDCOPY) |
| 16 | Mixed object document (MODCA) |
| 223 | Data file |

Also, refer to *Interchange Document Profile Reference* manual for more details. If this was a message-only distribution, this field contains zeros.

- *LINSYC* (Document system code). The document data is defined by this identifier. See the Send Distribution (SNDDST) command for further information. If this was a message-only distribution, this field contains blanks. This data is put into code from code page 256, character set 337.
- *LINCID* (Character set). The distribution description is put into code in this character set. This will be all zeros if no distribution description was supplied by the sender.
- *LINCPG* (Code page). The distribution description is put into code in this code page. This will be all zeros if no distribution description was supplied by the sender.
- *LINDSD* (Distribution description). This is 44 characters of text supplied by the sender of the distribution. This will be all blanks if no description was supplied. *LINCID* and *LINCPG* define the character set and code page of this data.
- *LINDEX* (Distribution identifier extension). This is a two-digit zoned numeric field that uniquely identifies duplicate distributions that were sent to a specific recipient. If a recipient receives N number of the same distributions, the distribution identifier extension for that recipient will be 1, 2, ... N. Therefore, for each distribution received, the distribution identifier is 1 and for each duplicate and thereafter, is incremented by 1.

- *LINMLS* (Mail log item status). This is a 1-character field that indicates the status of an incoming mail log item. The status values are:
 - 1 - New
 - 2 - Old unopened
 - 3 - Opened
 - 4 - Filed local
 - 5 - Filed remote
 - 6 - Filed deferred
 - 7 - Referenced library document deleted
 - 8 - Mail document destroyed or damaged
- *LINCTD* (Last date distribution changed). This is the date the document content was last changed. It is in the format YYYYMMDD, where:
 - YYYY = year
 - MM = month
 - DD = day
- *LINCTM* (Last time distribution changed). This is the time that the document content was last changed. It is in the format HHMNSS, where:
 - HH = hour
 - MN = minute
 - SS = second
- *LINCSY* (System name). The system on which the output file was created.
- *LINCDC* (Century). The century in which the output file was created (0 = 1900, 1 = 2000).
- *LINCDD* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *LINCDM* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *LINTCS* (Character set). The description is created in this character set. It is all zeros if no description meets the required criteria.

Note: Presentable characters must be greater than or equal to the EBCDIC representation for hex 40.
- *LINTCP* (Code page). The description is created in this code page. It is all zeros if no description meets the required criteria.

Note: Presentable characters must be greater than or equal to the EBCDIC representation for hex 40.
- *LINTXD* (Description). This is 44 characters of text supplied by the sender of the distribution. The description can be the following:
 - Subject
 - Document description
 - Distribution description

The description field takes one description if it meets the required criteria. If no supplied descriptors exist or if all contain characters that cannot be used, this field contains blanks. *LINTCS* and *LINTCP* define the character set and code page of this data.

Note: Presentable characters must be greater than or equal to the EBCDIC representation for hex 40.

- *LINCSX* (Character set). The distribution description is put into code in this character set. This is all zeros if no distribution description was supplied by the sender.
- *LINCPX* (Code page). The distribution description is put into code in this code page. This is all zeros if no distribution description was supplied by the sender.
- *LINNTE* (Note flag). A value of 1 indicates this is a note; 0 indicates anything other than a note.

Outgoing Distribution Information

The model output file supplied with the system is named QAOSILOT in library QSYS and is used for outgoing distribution information. One record is written to the file for each recipient on the distribution.

The following data description specifications were used to create the model output file:

```

A* Start of specifications
A*
A* File Name: QAOSILOT in library QSYS
A* File Type: Physical
A* Record Format Names: OSLOUT
A*
A*
A      R OSLOUT
A      OUTSNO      6S      TEXT('Sequence number')
A      OUTDID      20A     TEXT('Distribution +
A                          identifier')
A      OUTSUI      8A      TEXT('Sender identifier')
A      OUTSUA      8A      TEXT('Sender address')
A      OUTSDT      8A      TEXT('Send date')
A      OUTSTM      6A      TEXT('Send time')
A      OUTPER      1A      TEXT('Personal flag')
A      OUTPTY      1A      TEXT('Priority flag')
A      OUTCOD      1A      TEXT('Confirmation of +
A                          delivery flag')
A      OUTCID      3S      TEXT('Distribution +
A                          description character set')
A      OUTCPG      3S      TEXT('Distribution +
A                          description code page')
A      OUTDSD      44A     TEXT('Distribution +
A                          description')
A      OUTRUI      8A      TEXT('Recipient identifier')
A      OUTRUA      8A      TEXT('Recipient address')
A      OUTSTS      4S      TEXT('Status code')
A      OUTTYP      4S      TEXT('Status type')
A      OUTDAT      8A      TEXT('Status date')
A      OUTTIM      6A      TEXT('Status time')
A      OUTDEX      2S      TEXT('Distribution +
A                          identifier extension')
A      OUTCLS      1A      TEXT('Confirmation of
A                          delivery log status')
A      OUTLOG      1A      TEXT('Mail log type')
A      OUTCSY      8A      TEXT('System the outfile +
A                          was created on')
A      OUTCDC      1A      TEXT('Century the outfile +
A                          was created')
A      OUTCDD      6A      TEXT('Date the outfile +
A                          was created')
A      OUTCDM      6A      TEXT('Time the outfile +
A                          was created')
A      OUTCSX      5S      TEXT('Distribution +
A                          Document character set')
A      OUTCPX      5S      TEXT('Distribution +
A                          Document code page')
A      OUTNTE      1A      TEXT('Note flag')
A*
A* End of specifications

```

Figure C-27. QAOSILOT in Library QSYS Example

The meaning of each field in the record is:

- *OUTSNO* (Sequence number). A zoned numeric field that starts at 1 for the first record of a distribution and is incremented by 1 for the next record of a distribution. Note that this number continues to be incremented. It does not start over when the records of another distribution are written to the file.
- *OUTDID* (Distribution identifier). The unique identifier assigned to a distribution. It is composed of the user's address (8 characters), the user ID (8 characters), and a 4-byte zoned sequence number with leading zeros. For example: NEWYORK JACKSON 0023. Each record for a distribution has the same distribution ID. This data is put into code from code page 256, character set 930.
- *OUTSUI* (Sender's user ID). This is the user ID of the person who sent the distribution. This data is put into code from code page 256, character set 930.
- *OUTSUA* (Sender's address). This is the address of the person who sent the distribution. This data is put into code from code page 256, character set 930.
- *OUTSDT* (Date distribution sent). This is the date that the distribution was sent by the sender. It is in the format YYYYMMDD where:
 - YYYY = year
 - MM = month
 - DD = day
- *OUTSTM* (Time distribution sent). This is the time that the distribution was sent by the sender. It is in the format HHMNSS where:
 - HH = hour
 - MN = minute
 - SS = second
- *OUTPER* (Personal distribution). A value of 1 indicates this is a personal distribution; a value of 0 indicates it is not personal.
- *OUTPTY* (Distribution priority). A value of 1 indicates this was sent with high priority; a value of 0 indicates it was sent with normal priority.
- *OUTCOD* (Confirmation of delivery). A value of 1 indicates that the sender asked for confirmation of delivery. A value of 0 indicates that confirmation of delivery was not requested.
- *OUTCID* (Character set). The distribution description is put into code in this character set. This will be all zeros if no distribution description was supplied by the sender.
- *OUTCPG* (Code page). The distribution description is put into code in this code page. This will be all zeros if no distribution description was supplied by the sender.
- *OUTDSD* (Distribution description). This is 44 characters of text supplied by the sender of the distribution. This will be all blanks if no description was supplied. *OUTCID* and *OUTCPG* define the character set and code page of this data.
- *OUTRUI* (Recipient user ID). The user ID of a recipient or the first part of a distribution list ID. This data is put into code from code page 256, character set 930.

- *OUTRUA* (Recipient address). The user address of a recipient or the second part of a distribution list ID. This data is put into code from code page 256, character set 930.
- *OUTSTS* (Status code). A code that describes the current status of the distribution for this recipient. The following values are defined:

| Zoned Numeric Value | Description |
|---------------------|--|
| 0 | Distribution is in process |
| 1 | SNADS routing error |
| 2 | SNADS detected invalid user ID |
| 3 | SNADS hop count exceeded |
| 4 | SNADS detected syntax error |
| 5 | SNADS function not supported |
| 6 | SNADS permanent server error |
| 7 | SNADS unknown server name |
| 8 | SNADS invalid server parameters |
| 9 | SNADS invalid destination transaction program name |
| 12 | SNADS distribution purged |
| 13 | SNADS some user IDs lost |
| 14 | SNADS resource not available |
| 15 | SNADS system error |
| 16 | SNADS temporary server error |
| 17 | SNADS system I/O error |
| 18 | SNADS Acknowledge Interchange Unit error |
| 19 | SNADS distribution object size inconsistency error |
| 257 | DIA distribution was delivered |
| 513 | DIA a system error has occurred |
| 769 | DIA invalid local recipient |
| 1025 | DIA invalid document |
| 1281 | DIA recipient canceled the distribution |
| 1282 | DIA distribution was removed prior to delivery to the recipient |
| 1537 | DIA maximum list exceeded |
| 1793 | DIA some recipients canceled/purged and some delivered |
| 1794 | DIA some recipients delivered and some invalid |
| 1795 | DIA some recipients canceled/removed and some invalid |
| 1796 | DIA some recipients canceled/removed, some delivered, and some invalid |
| 1797 | DIA some recipients canceled and some removed |

Note: 17xx status codes apply when the recipient user ID was really a distribution list on a remote system. If all recipients on the remote distribution list have the same status, the common status code will be returned. For example, if all the recipients on the remote distribution list cancel the distribution, the status code will be 1281. But if the recipients on the remote distribution list have different status, one of the 17xx status codes will be returned for the status of the list.

- *OUTTYP* (Status type). The type of status that is set in the next field.
 - 0 - No status is set, distribution is in process
 - 1 - SNADS status
 - 512 - DIA status

- *OUTDAT* (Date status set). If the distribution has been delivered or error status has been returned, this is the date that this occurred. If the recipient is on another system in the office network, this is the date from that system. If no status has been returned yet, this field will contain zeros. It is in the format YYYYMMDD where:

YYYY = year
MM = month
DD = day

- *OUTTIM* (Time status set). If the distribution has been delivered or error status has been returned, this is the time that this occurred. If the recipient is on another system in the office network, this is the time from that system. If no status has been returned yet, this field will contain zeros. It is in the format HHMNSS where:

HH = hour
MN = minute
SS = second

- *OUTDEX* (Distribution identifier extension). This is a two-digit zoned numeric field that uniquely identifies duplicate distributions. This field will always be zero because there will never be a duplicate confirmation of delivery, error status delivery, or error status entry returned to the originator's mail log.

- *OUTCLS* (Confirmation of delivery log status). This is a 1-character field that indicates the status of an outgoing mail item or an undelivered or error mail item. The field values are:

0 = not applicable
1 = new
2 = old

- *OUTLOG* (Mail log type). This is a 1-character field that indicates the mail log type. The field values are:

C = confirmation of delivery log
U = undelivered or error status log

- *OUTCSY* (System name). The system on which the output file was created.

- *OUTCDC* (Century). The century in which the output file was created (0=1900, 1=2000).

- *OUTCDD* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.

- *OUTCDM* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.

- *OUTCSX* (Character set). The distribution description is put into code in this character set. This is all zeros if no distribution description was supplied by the sender.

- *OUTCPX* (Code page). The distribution description is put into code in this code page. This field is all zeros if no distribution description was supplied by the sender.

- *OUTNTE* (Note flag). A value of 1 indicates the original distribution was a note. A value of 0 indicates anything other than a note.

Receive Distribution (RCVDST) Output File Considerations

The Receive Distribution (RCVDST) command can be used to create a database file that contains a distribution sent to an office user. The distribution information as well as the distribution itself (for example, document or PC file) is written to the file.

The model output file supplied with the system is named QAOSIRCV in library QSYS.

The following data description specifications were used to create the model output file:

```

A* Start of specifications
A*
A* File Name: QAOSIRCV in library QSYS
A* File Type: Physical
A* Record Format Name: OSRCVD
A*
A*
A      R OSRCVD
A      RCVSNO      6S      TEXT('Sequence number')
A      RCVDID      20A     TEXT('Distribution +
A                               identifier')
A      RCVRUI      8A      TEXT('Recipient +
A                               identifier')
A      RCVRUA      8A      TEXT('Recipient address')
A      RCVSUI      8A      TEXT('Sender identifier')
A      RCVSUA      8A      TEXT('Sender address')
A      RCVSDT      8A      TEXT('Send date')
A      RCVSTM      6A      TEXT('Send time')
A      RCVPER      1A      TEXT('Personal flag')
A      RCVPTY      1A      TEXT('Priority flag')
A      RCVCOD      1A      TEXT('Confirmation of +
A                               delivery flag')
A      RCVDSZ      8S      TEXT('Document size')
A      RCVDTP      5S      TEXT('Document type')
A      RCVSYC      13A     TEXT('System code')
A      RCVCRS      3S      TEXT('Data character set')
A      RCVCPG      3S      TEXT('Data code page')
A      RCVDL       3S      TEXT('Data length')
A      RCVRC       3S      TEXT('Record code')
A      RCVDTA      500A    TEXT('Data')
A      RCVDEX      2S      TEXT('Distribution +
A                               identifier extension')
A      RCVCSY      8A      TEXT('System the outfile +
A                               was created on')
A      RCVCDC      1A      TEXT('Century the outfile +
A                               was created')
A      RCV added 6A      TEXT('Date the outfile +
A                               was created')
A      RCV added 6A      TEXT('Time the outfile +
A                               was created')
A      RCVCSX      5S      TEXT('Data character set')
A      RCVCPX      5S      TEXT('Data code page')
A*
A* End of specifications

```

Figure C-28. QAOSIRCV in Library QSYS

The meaning of each field in the record is:

- *RCVSNO* (Sequence number). A zoned numeric field that starts at 1 for the first record of a distribution and is incremented by 1 for the next record of a distribution.
- *RCVDID* (Distribution identifier). The unique identifier assigned to a distribution. It is composed of the user's address (8 characters), the user ID (8 characters), and a 4-byte zoned sequence number with leading zeros. For example: NEWYORK JACKSON 0023. Each record for a distribution has the same distribution ID. This data is put into code from code page 256, character set 930.
- *RCVRUI* (Recipient user ID). This is the user ID of the person who received the distribution. This data is put into code from code page 256, character set 930.
- *RCVRUA* (Recipient address). This is the address of the person who received the distribution. This data is put into code from code page 256, character set 930.
- *RCVSUI* (Sender's user ID). This is the user ID of the person who sent the distribution. This data is put into code from code page 256, character set 930.
- *RCVSUA* (Sender's address). This is the address of the person who sent the distribution. This data is put into code from code page 256, character set 930.
- *RCVSDT* (Date distribution sent). This is the date that the distribution was sent by the sender. If the sender is on another system in the office network, this is the date from that system. It is in the format YYYYMMDD where:
 - YYYY = year
 - MN = month
 - DD = day
- *RCVSTM* (Time distribution sent). This is the time that the distribution was sent by the sender. If the sender is on another system in the office network, this is the time from that system. It is in the format HHMNSS where:
 - HH = hour
 - MN = minute
 - SS = second
- *RCVPER* (Personal distribution). A value of 1 indicates this is a personal distribution; 0 indicates it is not personal.
- *RCVPTY* (Distribution priority). A value of 1 indicates this was sent with high priority; 0 indicates it was sent with normal priority.
- *RCVCOD* (Confirmation of delivery). A value of 1 indicates that the sender asked for confirmation of delivery. A value of 0 indicates that confirmation of delivery was not requested.
- *RCVDSZ* (Document size). This is the size in bytes of the document data. To get the size of the IDP in the document, take the sum of all the QDLDL fields when the field QDLRC is equal to record code 500.

- *RCVDTP* (Document type). The document data is defined by this identifier.

| DOCUMENT TYPE VALUE | DOCUMENT TYPE MEANING |
|---------------------|---|
| 2 | Final-form text document (FFTDCA) |
| 3 | 5520 revisable-form text document |
| 4 | Word-processing EBCDIC |
| 5 | Word-processing information file (WPIF) |
| 6 | Image-data subset document |
| 7 | 3730 text data stream |
| 8 | DIA document library document descriptor document |
| 9 | 3732 display document data stream |
| 10 | DIA defined document unit content |
| 11 | Revisable-form text document (RFTDCA) |
| 12 | 1403 printer compatible data stream with variable length, unblocked records |
| 13 | Digitized ADS audio |
| 14 | IBM PC data file |
| 15 | Printed copy of document |
| 16 | Mixed object document |
| 223 | Data file |

Also, refer to the *Interchange Document Profile Reference* manual for more details. If this was a message-only distribution, this field contains zeros.

- *RCVSYC* (Document system code). The document data is defined by this identifier. This data is put into code from code page 256, character set 930. See the Send Distribution (SNDDST) command for further information.
- *RCVCRS* (Character set). The data field (*RCVDTA*) is put into code in this character set.
- *RCVCPG* (Code page). The data field (*RCVDTA*) is put into code in this code page.
- *RCVDL* (Data length). This is the length of data in the data field (*RCVDTA*) for this record.
- *RCVRC* (Record code). This is the record code for this record. The content of the following field is determined by this code.
- *RCVDTA* (Data field). The contents of this field are described below.

| Record Code | Data Field | Description |
|-------------|------------|---|
| 010 | RCVDTA | Distribution description. This is 44 characters of text supplied by the sender of the distribution. <i>RCVCRS</i> and <i>RCVCPG</i> define the character set and code page of this data. |
| 020 | RCVDTA | Message. This is a 1- to 256-character message associated with the distribution. <i>RCVCRS</i> and <i>RCVCPG</i> define the character set and code page of this data. |
| 105 | RCVDTA | Document description. This is a 1- to 44-character name assigned to the document. The system applies no meaning to this description. <i>RCVCRS</i> and <i>RCVCPG</i> define the character set and code page of this data. |

| Record Code | Data Field | Description |
|--------------------|-------------------|---|
| 110 | RCVDTA | <p>Creation date and time. The date and time the document was created. This date and time are contained in 12 characters in the format YYYYMMDDHHMN where:</p> <p>YYYY = year MM = month DD = day HH = hour MN = minute</p> |
| 115 | RCVDTA | <p>Expiration date. This is the date when the document is no longer needed. This date is contained in 8 characters in the format YYYYMMDD where:</p> <p>YYYY = year MM = month DD = day</p> |
| 120 | RCVDTA | <p>Document date. This is any unique date the user wishes to associate with the document. This date is contained in 8 characters in the format YYYYMMDD where:</p> <p>YYYY = year MM = month DD = day</p> |
| 125 | RCVDTA | <p>File date and time. This is the date and time the document was filed in the system's document library. This date and time were automatically created by the system when the document was filed. This date and time are contained in 16 characters in the format YYYYMMDDHHMNSSHS where:</p> <p>YYYY = year MM = month DD = day HH = hour MN = minute SS = second HS = hundredths of second</p> |
| 130 | RCVDTA | <p>Revision Date and time. This is the date and time the document was last changed. This date and time are contained in 16 characters in the format YYYYMMDDHHMNSSHS where:</p> <p>YYYY = year MM = month DD = day HH = hour MN = minute SS = second HS = hundredths of second</p> |
| 135 | RCVDTA | <p>Action due date. This is the date when the action specified in the document is due. This date is contained in 8 characters in the format YYYYMMDD where:</p> <p>YYYY = year MM = month DD = day</p> |

| Record Code | Data Field | Description |
|-------------|------------|--|
| 140 | RCVDTA | Completion date. This is the date when the action specified in the document is complete. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |
| 145 | RCVDTA | Author name. This is the name of the author of the document (1 to 20 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 150 | RCVDTA | Copy name. This is the name of one entry in the copy list (1 to 60 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 155 | RCVDTA | Document class. This is the document class (1 to 16 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 160 | RCVDTA | File cabinet reference. This refers to offline storage where a printed document resides (1 to 60 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 165 | RCVDTA | Subject. This is the subject of the document (1 to 60 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 170 | RCVDTA | Keyword. This is a keyword associated with the document (1 to 35 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 175 | RCVDTA | Reference. This is the reference associated with the document (1 to 60 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 180 | RCVDTA | Status. This is the user-defined work progress descriptor associated with the document (1 to 20 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 185 | RCVDTA | Project. This is the user-defined work identifier associated with the document (1 to 10 characters). RCVCRS and RCVCPG define the character set and code page of this data. |
| 500 | RCVDTA | IDP data. This is the interchange document profile (IDP) that was distributed with the document. It conforms to the data stream definition of the Document Interchange Architecture. Refer to the <i>Interchange Document Profile Reference</i> manual for details. This data is optionally provided for the user that needs to access the IDP for fields other than those provided by separate record codes. It is the user's responsibility to parse this data and extract the desired information. This field will be 500 bytes of the IDP per record. The last record with this record code can have less than 500 bytes. The data length is defined by RCVDL. |

| Record Code | Data Field | Description |
|-------------|------------|---|
| 800 | RCVDTA | Document data. This is the data that was distributed and is not defined by the CL command. The meaning of the data stream is defined by the document type and the system code. This field will contain 500 bytes of the document per record. The last record with this record code can have less than 500 bytes. The data length is defined by RCVDL. RCVCRS and RCVCPG define the character set and code page of this data. RFTDCA and FFTDCA data streams are described in the <i>Revisable-Form Text Reference</i> and <i>Final-Form Text Reference</i> manuals. |

- *RCVDEX* (Distribution identifier extension). This is a two-digit zoned numeric field that uniquely identifies duplicate distributions that were sent to a specific recipient. If a recipient receives N number of the same distributions, the distribution identifier extension for that recipient will be 1, 2, ... N. Therefore, for each distribution received, the distribution identifier is 1, and for each duplicate thereafter, is incremented by 1.
- *RCVCSY* (System name). The system on which the output file was created.
- *RCVDCD* (Century). The century in which the output file was created (0=1900, 1=2000).
- *RCVCDD* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *RCVCDM* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *RCVCSX* (Character set). The data field (RCVDTA) is put into code in this character set.
- *RCVCPX* (Code page). The data field (RCVDTA) is put into code in this code page.

Replace Document (RPLDOC) File Considerations

The Replace Document (RPLDOC) command accepts a database file as input. The file can be a source file or data file with no office considerations. In this case, the data is read and replaces a document in the document library.

The file can also be a data file that is formatted to be the same as the output file that the Retrieve Document (RTVDOC) command creates or the file can be a data file that is formatted to be the same as the output file that the Receive Distribution (RCVDST) command creates. When the file is in this form, the RPLDOC command will recognize that and properly process the file to format the document content and the interchange document profile (IDP).

Thus it is possible to issue a Retrieve Document command to copy a document to a file and later file that document back in the document library using the Replace Document command or to receive a distribution and use the output file of that command to replace a document in the document library. It is also possible to write your own application to format a data file in the same manner as the Retrieve Document command does and use that file as input to the Replace Document command.

Retrieve Document (RTVDOC) Output File Considerations

The Retrieve Document (RTVDOC) command can be used to create a database file that contains the document and document profile information that has been copied from the document library.

The model output file supplied with the system is named QAOSIRTV in library QSYS.

The following data description specifications were used to create the model output file:

```
A* Start of specifications
A*
A* File Name: QAOSIRTV in library QSYS
A* File Type: Physical
A* Record Format Name: OSRTVD
A*
A*
A      R OSRTVD
A      RTVSNO      6S      TEXT('Sequence number')
A      RTVDID      24A     TEXT('Document +
A                          identifier')
A      RTVDNM      12A     TEXT('Document name')
A      RTVFLR      63A     TEXT('Folder name')
A      RTVONM      10A     TEXT('Object name')
A      RTVUID      8A      TEXT('Owner user ID')
A      RTVADR      8A      TEXT('Owner address')
A      RTVDSZ      8S      TEXT('Document size')
A      RTVDTP      5S      TEXT('Document type')
A      RTVSYC      13A     TEXT('System code')
A      RTVACF      1A      TEXT('Access flag')
A      RTVPER      1A      TEXT('Personal flag')
A      RTVWIP      1A      TEXT('Work in process +
A                          flag')
A      RTVAHV      1A      TEXT('Mark for offline flag')
A      RTVCKO      1A      TEXT('Checkout flag')
A      RTVCRS      3S      TEXT('Data character set')
A      RTVCPG      3S      TEXT('Data code page')
A      RTVDL      3S      TEXT('Data length')
A      RTVRC      3S      TEXT('Record code')
A      RTVDTA      500A    TEXT('Data')
A      RTVCSY      8A      TEXT('System the outfile +
A                          was created on')
A      RTVCDC      1A      TEXT('Century the outfile +
A                          was created')
A      RTVCDD      6A      TEXT('Date the outfile +
A                          was created')
A      RTVCDM      6A      TEXT('Time the outfile +
A                          was created')
A      RTVCSX      5S      TEXT('Data character set')
A      RTVCPX      5S      TEXT('Data code page')
A*
A* End of specifications
```

Figure C-29. QAOSIRTV in Library QSYS Example

| DOCUMENT TYPE VALUE | DOCUMENT TYPE MEANING |
|---------------------|---|
| 2 | Final-form text document (FFTDCA) |
| 3 | 5520 revisable-form text document (RFT5520) |
| 4 | Word-processing EBCDIC (EBCDIC) |
| 5 | Word-processing information file (WRDPRCINF) |
| 6 | Image-data subset document (IMAGE) |
| 7 | 3730 text data stream (TEXT3730) |
| 8 | DIA document library document descriptor document (DOCDESCDOC) |
| 9 | 3732 display document data stream (DATAST3732) |
| 10 | DIA defined document unit content (DOCUNITCTL) |
| 11 | Revisable-form text document (RFTDCA) |
| 12 | 1403 printer compatible data stream with variable length, unblocked records (PRT1403) |
| 13 | Digitized ADS audio (AUDIO) |
| 14 | IBM PC data file (PCFILE) |
| 15 | Printed copy of document (HARDCOPY) |
| 16 | Mixed object document (MODCA) |
| 223 | Data file |

Also, refer to the *Interchange Document Profile Reference* manual for more details.

- **RTVSYC** (Document system code). The document data is defined by this identifier. See the File Document (FILDOC) command for further information. This data is put into code from code page 256, character set 337.
- **RTVACF** (Access flag). A 1-character flag to indicate if the document is a private or public document.
 - 0 =public
 - 1 =private
- **RTVPER** (Personal flag). A 1-character flag to indicate if the document is personal (not available to affinity users).
 - 0 =not personal
 - 1 =personal
- **RTVWIP** (Work in process flag). A 1-character flag to indicate a revisable (work in process) or a final (non-replaceable) document.
 - 0 =final
 - 1 =revisable
- **RTVAHV** (Mark for offline flag). A 1-character flag to indicate if a document has been marked for saving to offline storage.
 - 0 =not marked for save
 - 1 =marked for save only
 - 2 =marked for save and delete document object
 - 3 =marked for save and delete all data
- **RTVCKO** (Checkout flag). A 1-character flag to indicate if a revisable document has been checked out for update.
 - 0 =not checked out
 - 1 =checked out for update
- **RTVCRS** (Data field character set). The data in the data field is in this character set. This field is set to 999 if the character set is greater than 999. **RTVCSX** can be used to get the data field character set.

- *RTVCPG* (Data field code page). The data in the data field is in this code page. This field is set to 999 if the code page is greater than 999. *RTVCPX* can be used to get the data field code page.
- *RTVDL* (Length of data). This number defines the length of the data field. It encompasses only the data field. For some records, the length is not really necessary because the following data is a fixed length. For variable length data it is necessary. It is correctly set for all records.
- *RTVRC* (Record code). A unique zoned number is defined for each record type. The record code must be used to determine the definition of the data to follow.
- *RTVDTA* (Data field). This field takes on many meanings and is further described in the following table.

| Record Code | Data Field | Description |
|-------------|------------|--|
| 105 | RTVDTA | Document description. This is a 1- to 44-character document name assigned by the user. The system applies no meaning to this description. <i>RTVCRS</i> and <i>RTVCPG</i> define the character set and code page of this data. |
| 110 | RTVDTA | Creation date and time. This is the date and time the document was created. This date and time are contained in 12 characters in the format YYYYMMDDHHMN where: YYYY = year MM = month DD = day HH = hour MN = minute |
| 115 | RTVDTA | Expiration date. This is the date when the document is no longer needed. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |
| 120 | RTVDTA | Document date. This is any unique date the user wishes to associate with the document. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |
| 125 | RTVDTA | File date and time. This is the date and time the document was filed in the system's document library. This date and time were automatically created by the system when the document was filed. This date and time are contained in 16 characters in the format YYYYMMDDHHMNSSHS where: YYYY = year MM = month DD = day HH = hour MN = minute SS = second HS = hundredths of second |

| Record Code | Data Field | Description |
|--------------------|-------------------|--|
| 130 | RTVDTA | Revision date and time. This is the date and time the document content or document details were last changed. This date and time are contained in 16 characters in the format YYYYMMDDHHMNSSHS where: YYYY = year MM = month DD = day HH = hour MN = minute SS = second HS = hundredths of second |
| 135 | RTVDTA | Action due date. This is the date when the action specified in the document is due. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |
| 140 | RTVDTA | Completion date. This is the date when the action specified in the document is complete. This date is contained in 8 characters in the format YYYYMMDD where: YYYY = year MM = month DD = day |
| 145 | RTVDTA | Author name. This is the name of the author of the document (1 to 20 characters). RTVCRS and RTVCPG define the character set and code page of this data. |
| 150 | RTVDTA | Copy name. This is the name of one entry in the copy list (1 to 60 characters). RTVCRS and RTVCPG define the character set and code page of this data. |
| 155 | RTVDTA | Document class. This is the document class (1 to 16 characters). |
| 160 | RTVDTA | File cabinet reference. This contains any reference to offline storage where a printed document resides (1 to 60 characters). RTVCRS and RTVCPG define the character set and code page of this data. |
| 165 | RTVDTA | Subject. This is the subject of the document (1 to 60 characters), the character set and code page apply. |
| 170 | RTVDTA | Keyword. This is a keyword associated with the document (1 to 35 characters). RTVCRS and RTVCPG define the character set and code page of this data. |
| 175 | RTVDTA | Reference. This is the reference associated with the document (1 to 60 characters). RTVCRS and RTVCPG define the character set and code page of this data. |
| 180 | RTVDTA | Status. This is the user-defined work progress descriptor associated with the document (1 to 20 characters). RTVCRS and RTVCPG define the character set and code page of this data. |

| Record Code | Data Field | Description |
|-------------|------------|--|
| 185 | RTVDTA | Project. This is the user-defined work identifier associated with the document (1 to 10 characters). RTVCRS and RTVCPG define the character set and code page of this data. |
| 500 | RTVDTA | Interchange document profile (IDP) details. These are the document details that are kept with the document. It conforms to the data stream definition of the Document Interchange Architecture. Refer to the <i>Interchange Document Profile Reference</i> manual for details. This data is optionally provided for the user that needs to access the IDP for fields other than those provided by separate record codes. It is the user's responsibility to parse this data and extract the desired information. This field will be 500 bytes of the IDP per record. The last record with this record code can have less than 500 bytes. The data length is defined by RTVDL. RTVCRS and RTVCPG define the character set and code page of this data. |
| 800 | RTVDTA | Document content. This is the content that was distributed and is not defined by the CL command. The meaning of the data stream is defined by the document type and the system code. This field will contain 500 bytes of the document per record. The last record with this record code can have less than 500 bytes. The data length is defined by RTVDL. FFTDCA and RFTDCA data streams are described in the <i>Revisable-Form Text Reference</i> and <i>Final-Form Text Reference</i> manuals. |

- *RTVCSY* (System name). The system on which the output file was created.
- *RTVCDC* (Century). The century in which the output file was created (0=1900, 1=2000).
- *RTVCDD* (Date). The date the output file was created. The date is specified as YYMMDD, where YY=year, MM=month, and DD=day.
- *RTVCDM* (Time). The time the output file was created. The time is specified as HHMMSS, where HH=hour, MM=minutes, and SS=seconds.
- *RTVCSX* (Data field character set). The data in the data field is in this character set.
- *RTVCPX* (Data field code page). The data in the data field is in this code page.

Save Document Library Object (SAVDLO) Output File Considerations

The Save Document Library Object (SAVDLO) command provides the OUTPUT(*OUTFILE) option to create a database file that contains information about the documents and folders that are saved. The information in the database file is the same information that would be displayed in a listing if the OUTPUT(*PRINT) option were chosen.

Folder Object Information

The model output file supplied with the system is named QAOJSAVO in library QSYS.

The following data description specifications were used to create the model output file:

```
A* Start of specifications
A*
A* File Name: QAOJSAVO in library QSYS
A* File Type: Physical
A* Record Format Name: OJSDLO
A*
A*
A      R OJSDLO
A      SDLTYP          2S      TEXT('Record type')
A      SDLSDT         12A      TEXT('Save date/time')
A      SDLDEV         10A      TEXT('Device')
A      SDLLBL         17A      TEXT('Label')
A      SDLSEQ          4A      TEXT('Sequence number')
A      SDLVID         90A      TEXT('Volume ID(s)')
A      SDLPPTH        63A      TEXT('Folder name')
A      SDLDOC         20A      TEXT('Document name')
A      SDLDSC         44A      TEXT('Document
                                description')
A      SDLCDT          6A      TEXT('Creation date/
                                time')
A      SDLOWN         10A      TEXT('Owner name')
A      SDLSON         10A      TEXT('Document system
                                Object name')
A      SDLSID          8A      TEXT('Sender's user
                                id')
A      SDLSAD          8A      TEXT('Sender's
                                address')
A      SDLDST         44A      TEXT('Distribution
                                description')
A      SDLDTS          6A      TEXT('Date sent')
A      SDLCSY          8A      TEXT('System Name')
A      SDLSFN         10A      TEXT('Save file name')
A      SDLSFL         10A      TEXT('Save file +
                                library')
A      SDSDTC          1A      TEXT('SAVDLO century - +
                                0=19xx 1=20xx')
```

Figure C-30 (Part 1 of 2). QAOJSAVO in Library QSYS Example

| | | | |
|---|--------|------|--|
| A | SDCDTC | 1A | TEXT('Document creation + century - 0=19xx + 1=20xx') |
| A | SDDTSC | 1A | TEXT('Document sent + century - 0=19xx + 1=20xx') |
| A | SDLFRE | 1A | TEXT('Storage freed - + 1=*FREE, 0=*KEEP, + 2=*DELETE') |
| A | SDLCPR | 1A | TEXT('Data compressed + - 0=*NO, + 1=*YES') |
| A | SDLCPC | 1A | TEXT('Data compacted - + 0=*NO', + 1=*YES') |
| A | SDLTGT | 6A | TEXT('Target release - + version, release, + modification - VvRrMm') |
| A | SDSTAT | 1A | TEXT('Status - 1=Saved, + 0=Not saved') |
| A | SDMGID | 7A | TEXT('Message ID for obj+ ects not saved') |
| A | SDLVI2 | 360A | TEXT('Volume IDs for + volumes 16 - 75') VARLEN |

A*
A* End of specifications

Figure C-30 (Part 2 of 2). QAOJSAVO in Library QSYS Example

The meaning of each field in the record is:

- **SDLTYP** (Record type). The record type 01 indicates the record is for a folder. The record type 02 indicates the record is for a filed document that may or may not be in a folder. The record type 03 indicates the record is for an unfiled document.
- **SDLSDT** (Save date/time). The date and time that the documents and folders were saved.
- **SDLDEV** (Device). The device used for the SAVDLO.
- **SDLLBL** (Label). The label applies to diskettes or tapes and is the data file identifier for the diskette file or tape file that contains the saved documents or folders.
- **SDLSEQ** (Sequence number). The sequence number of the tape file contains the saved documents and folders.
- **SDLVID** (Volume ID). The volume IDs (up to 15) of the diskettes or tapes that contain the saved documents and folders.
- **SDLPTH** (Folder). The fully qualified folder name of a folder that was saved (record type 01).
- **SDLDOC** (Document name). The user-specified name of a filed document that was saved (record type 02).
- **SDLDSC** (Document description). The document description of filed documents (record type 02).

- *SDLCDT* (Creation date/time). The date/time the filed document was created (record type 02).
- *SDLOWN* (Owner name). The owner of the document or folder that was saved.
- *SDLSON* (Document system-object name). The system-object name of the filed document (record type 02).
- *SSDSID* (Sender's user ID). The user ID of the person who sent the unfiled document (record type 03).
- *SDLSAD* (Sender's address). The address of the person who sent the unfiled document (record type 03).
- *SDL DST* (Distribution description). The description of the distribution for the saved unfiled document (record type 03).
- *SDL DTS* (Date sent). The date the saved unfiled document was sent (record type 03).
- *SDLC SY* (System name). The name of the system from which the document or folder was saved.
- *SDLSFN* (Save file name). The name of the save file used to save the documents and folders. This field is valid only when DEV(*SAVF) is specified.
- *SDLSFL* (Save file library). The name of the library containing the save file used to save the documents and folders. This field is valid only when DEV(*SAVF) is specified.
- *SDSDTC* (SAVDLO century). This field contains a 1-character value to show the century the document was saved (9=20th century, 0=21st century). Combine this field with *SDLSDT* to form a 13-digit date and time.
- *SDCDTC* (Document creation century). This field contains a 1-character value to show the century the document was created (9=20th century, 0=21st century). Combine this field with *SDLCDT* to form a 7-digit date.
- *SDDTSC* (Document sent century). This field contains a 1-character value to show the century the document was sent (9=20th century, 0=21st century). Combine this field with *SDL DTS* to form a seven-digit date.
- *SDLFRE* (Storage freed indicator). This field contains a 1-character indicator that shows the value of the STG parameter (F=*FREE, K=*KEEP, D=*DELETE).
- *SDLCPR* (Data compressed). This field contains a 1-character indicator that shows if the data was compressed (Y=compressed, N=not compressed).
- *SDLCPC* (Data compacted). This field contains a 1-character indicator that shows if the data was compacted (Y=compacted, N=not compacted).
- *SDLTGT* (Target release). The release to which the output from this SAVDLO request is intended to be restored.
- *SDSTAT* (Save status). This field contains a 1-character indicator that shows the save status of the document (Y=saved, N=not saved).
- *SDMGID* (Message ID for object not saved.)
- *SDLV12* (Volumes IDs). The 16th through the 75th volume IDs of the diskettes or tapes used to save the documents and folders. This is a variable length

field. To access this data, programs must be compiled to access variable length fields.

Send Distribution (SNDDST) Output File Considerations

The Send Distribution (SNDDST) command accepts a database file as input. The file can be a source file or data file with no office considerations. In this case, the data is read and sent within the office network.

The file can also be a data file that is formatted to be the same as the output file that the Receive Distribution (RCVDST) command creates or the file can be a data file that is formatted to be the same as the output file that the Retrieve Document (RTVDOC) command creates. When the file is in this form, the Send Distribution command will recognize that and properly process the file to format the data and the interchange document profile (IDP).

Thus it is possible to issue a Receive Distribution command to put a distribution in a file and send that distribution to someone using the Send Distribution command. Or you can use the Retrieve Document command to copy a document from the document library into a database file and then later send it to someone using the Send Distribution command. It is also possible to write your own application to format a data file in the same manner as the Receive Distribution command does and use that file as input to the Send Distribution command.

Appendix D. Control Language (CL) Commands

Following are the commands that are part of the OfficeVision/400 licensed program and are described in this manual. You must have the OfficeVision/400 licensed program installed to use these commands.

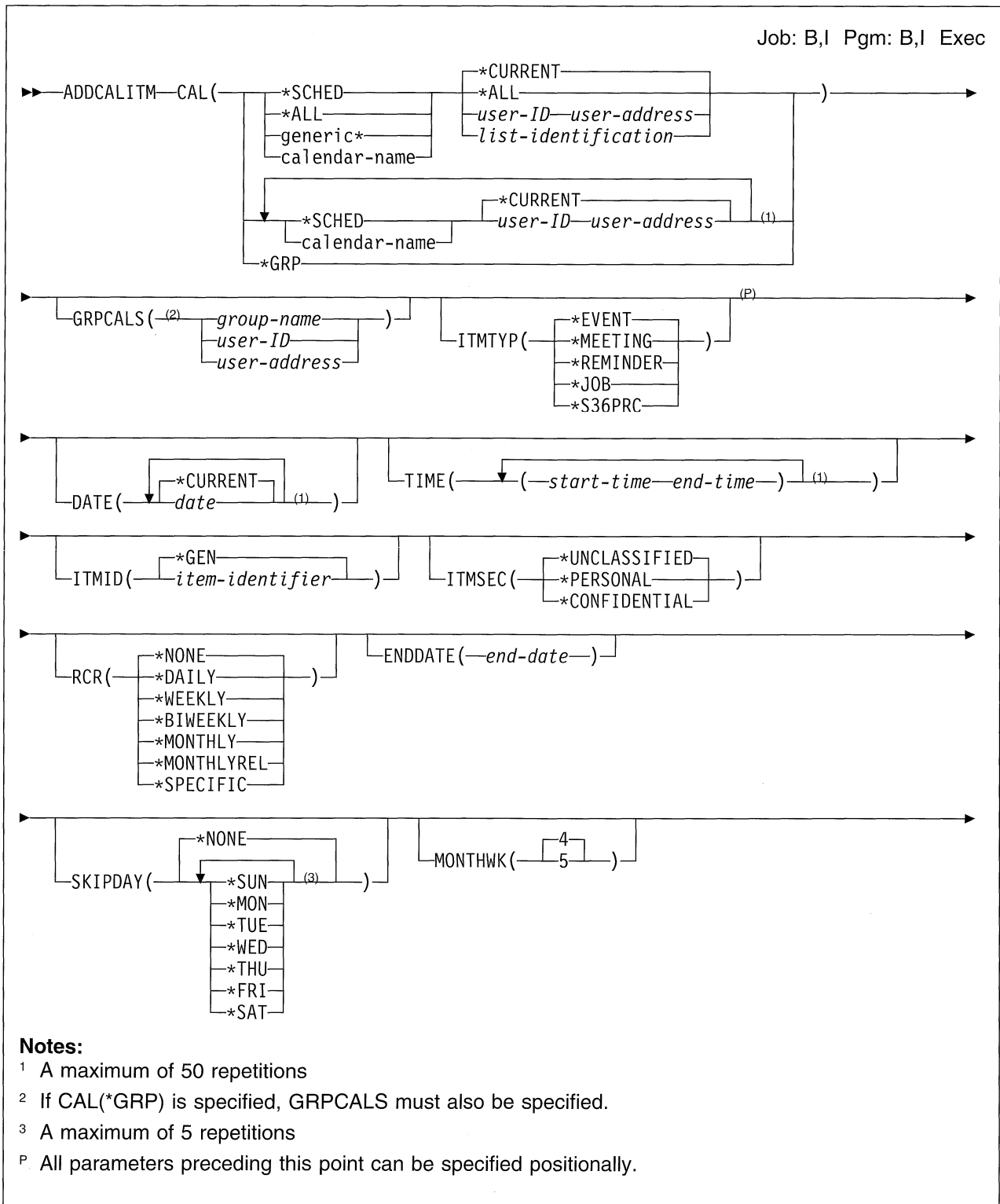
- “ADDCALITM (Add Calendar Item) Command” on page D-4
- “ADDDOCCVN (Add Document Conversion) Command” on page D-12
- “ADDOFCENR (Add Office Enrollment) Command” on page D-15
- “ADDOFCAPPE (Add Office Application Entry) Command” on page D-14
- “ADDTXTIDX (Add Text Index Entry) Command” on page D-20
- “CHGCALAUT (Change Calendar Authority) Command” on page D-23
- “CHGDOCCVN (Change Document Conversion) Command” on page D-28
- “CHGDOCSET (Change Document Set) Command” on page D-30
- “CHGDOCTYP (Change Document Type) Command” on page D-31
- “CHGOFCA (Change Office Attributes) Command” on page D-33
- “CHGOFCAPPD (Change Office Application Description) Command” on page D-34
- “CHGOFCAPPE (Change Office Application Entry) Command” on page D-35
- “CHGOFCENR (Change Office Enrollment) Command” on page D-36
- “CHKDOC (Check Document) Command” on page D-42
- “CRTCAL (Create Calendar) Command” on page D-44
- “CRTDOCSET (Create Document Set) Command” on page D-50
- “CRTDOCTYP (Create Document Type) Command” on page D-51
- “CRTOFCAPPD (Create Office Application Description) Command” on page D-54
- “CVTDOC (Convert Document) Command” on page D-55
- “DLTCAL (Delete Calendar) Command” on page D-56
- “DLTDOCSET (Delete Document Set) Command” on page D-58
- “DLTDOCTYP (Delete Document Type) Command” on page D-59
- “DLTOFCAPPD (Delete Office Application Description) Command” on page D-60
- “DLTOUTMAIL (Delete Outgoing Mail) Command” on page D-61
- “DSPCALAUT (Display Calendar Authority) Command” on page D-62
- “DSPCALD (Display Calendar Description) Command” on page D-65
- “DSPCALITM (Display Calendar Item) Command” on page D-68
- “DSPDOCCVN (Display Document Conversion) Command” on page D-71
- “DSPIDXSTS (Display Text Index Status) Command” on page D-73
- “ENDCALSRV (End Calendar Service) Command” on page D-74
- “ENDIDXMON (End Index Monitor) Command” on page D-75
- “FILLFORM (Fill Form Document) Command” on page D-76
- “MRGDOC (Merge Document) Command” on page D-79
- “PAGDOC (Paginate Document) Command” on page D-84
- “PRTCAL (Print Calendar) Command” on page D-85
- “QRYCALITM (Query Calendar Item) Command” on page D-94
- “RMVCALITM (Remove Calendar Item) Command” on page D-104
- “RMVDOCCVN (Remove Document Conversion) Command” on page D-107
- “RMVOFCENR (Remove Office Enrollment) Command” on page D-109
- “RMVOFCAPPE (Remove Office Application Entry) Command” on page D-108
- “RMVTEXTIDX (Remove Text Index Entry) Command” on page D-110
- “RSTCAL (Restore Calendar) Command” on page D-113
- “RTVOFCA (Retrieve Office Attributes) Command” on page D-117
- “RTVOFCENR (Retrieve Office Enrollment) Command” on page D-119
- “SAVCAL (Save Calendar) Command” on page D-122
- “SNDDOC (Send Document) Command” on page D-127
- “STRCALSRV (Start Calendar Service) Command” on page D-128

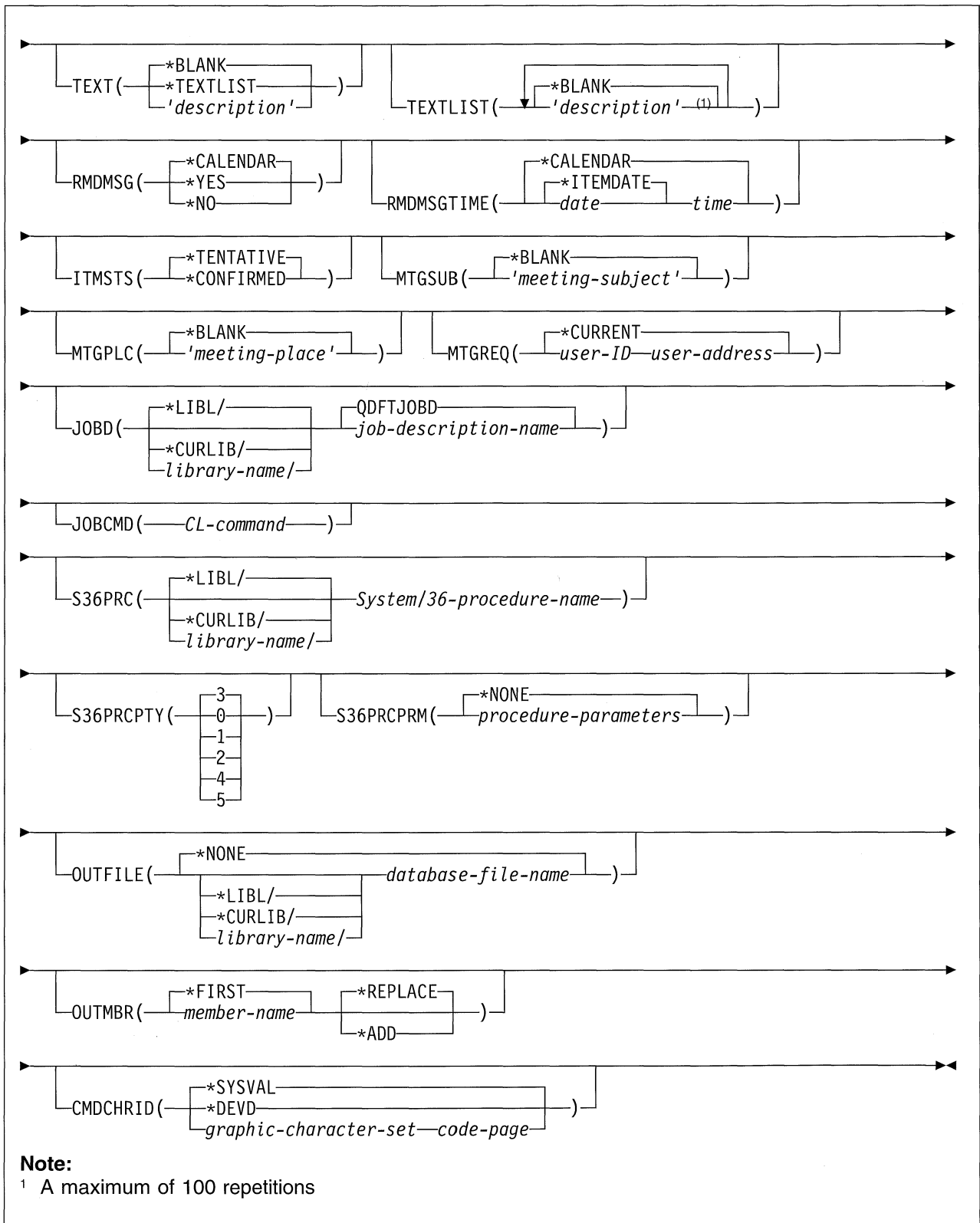
- “STRIDXMN (Start Index Monitor) Command” on page D-129
- “STROFC (Start Office) Command” on page D-130
- “STRPRSDIR (Start Personal Directory) Command” on page D-131
- “STRRGZIDX (Start Reorganization of Index) Command” on page D-132
- “STRUPDIDX (Start Update of Index) Command” on page D-133
- “STRWP (Start Word Processing) Command” on page D-134
- “WRKDOCCVN (Work with Document Conversions) Command” on page D-135
- “WRKDOCSET (Work with Document Sets) Command” on page D-136
- “WRKDOCTYP (Work with Document Types) Command” on page D-137
- “WRKOFCAPPD (Work with Office Application Descriptions) Command” on page D-138
- “WRKOFCAPPE (Work with Office Application Entries) Command” on page D-139
- “WRKTXIDX (Work with Text Index) Command” on page D-140
- “WRKTXTPRF (Work with Text Profiles) Command” on page D-141

The following commands are not part of the OfficeVision/400 licensed program and are described in the *CL Reference* manual and the *Programming Reference Summary*.

| | | |
|-----------|------------|------------|
| ADDDIRE | DLTSPADCT | RMVDIRSHD |
| ADDDIRSHD | DMPDLO | RMVDLOAUT |
| ADDLOAUT | DSPAUTLDLO | RMVDSTLE |
| ADDSTLE | DSPDIR | RNMDIRE |
| CFGRPDS | DSPDLOAUD | RNMDLO |
| CHGDIRE | DSPDLOAUT | RTVDLONAM |
| CHGDIRSHD | DSPDLONAM | RPLDOC |
| CHGDLOAUD | DSPDOC | RSTDLO |
| CHGDLOAUT | DSPDSTL | RTVDOC |
| CHGDLOOWN | DSPFLR | RVKUSRPMN |
| CHGDOCD | DSPHLPDOC | SAVDLO |
| CHGDSTD | DSPUSRPMN | SNDDST |
| CHKDLO | EDTDLOAUT | WRKDIR |
| CPYDOC | EDTDOC | WRKDIRLOC |
| CPYFRMDIR | FILDOC | WRKDIRSHD |
| CPYFRMPCD | GRTUSRPMN | WRKDOC |
| CPYTODIR | MOVDOC | WRKDOCLIB |
| CPYTOPCD | OVRPRTF | WRKDOCPRTO |
| CRTDOC | PRTDOC | WRKDSTL |
| CRTDSTL | QRYDOCLIB | WRKFLR |
| CRTFLR | QRYDST | |
| CRTSPADCT | RCLDLO | |
| DLTDLO | RCVDST | |
| DLTDST | RGZDLO | |
| DLTDSTL | RMVDIRE | |

ADDCALITM (Add Calendar Item) Command





Purpose

The Add Calendar Item (ADDCALITM) command is used to add calendar items to local calendars. If the item is a meeting, remote calendars are allowed and a meeting notice can be sent. A meeting notice is also sent when the scheduler is not authorized to add items to calendars of invited users.

Note: If you do not have the mail option installed and distributions are not allowed, meeting notices cannot be sent.

Restriction: The user of this command must be in the system distribution directory.

Parameters

CAL

Specifies the name and owner of the calendar or calendars to which calendar items are added. Up to 50 calendar names and calendar owners or *GRP can be specified on this parameter.

Element 1: Calendar Name

***SCHED:** The item is added to the scheduling calendars owned by the users specified on Element 2 of this parameter.

***ALL:** The item is added to all calendars owned by the users specified on Element 2 of this parameter.

calendar-name: Specify the name of the calendar to which the item is added.

generic-name:* Specify the generic name of the calendar. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then items are added to all calendars with names that begin with the generic name, and for which the user has authority. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete calendar name.

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify "MIKE**" to add the item to all calen-

dars that have names beginning with "MIKE**".

Element 2: Calendar Owner

***CURRENT:** The item is added to the specified calendars owned by the user issuing the command.

***ALL:** The item is added to all calendars that have names specified on Element 1 and are owned by local users.

list-identification: Specify a distribution list. The item is added to all calendars that have names specified on Element 1 and are owned by users on the distribution list.

user-ID: Specify the user ID of the owner of the calendar to which the item is added. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar to which the item is added. This value is valid only if user ID is also specified.

Other Single Values:

***GRP:** The item is added to all calendars in the calendar group specified on the GRPCALS parameter.

GRPCALS

Specifies the local calendar group to which the item is added. This parameter is valid only if CAL(*GRP) is also specified. A value for Element 1 must be specified if GRPCALS is specified.

Element 1: Group Name

group-name: Specify the local calendar group to which the item is added.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group.

Element 3: User Address

user-address: Specify the user address of the owner of the calendar group.

ITMTYP

Specifies the type of item that is added to the calendar.

***EVENT:** An event is added to the calendar. An event is an item that has a starting time and an ending time and is added to only one

calendar. The same event can be on several calendars, but events (unlike meetings) are not related.

***MEETING:** A meeting is added to the calendar. A meeting is an item that has a starting time and an ending time and is added to more than one calendar. The system provides methods to work with meeting items, indicate who is attending a meeting, and create meeting entries on the calendars of users invited to a meeting. If the user issuing the command is not authorized to an invited user's calendar, the invited user receives a meeting notice.

***REMINDER:** A reminder is added to the calendar. A reminder is associated with a day and does not take up any time on the calendar.

***JOB:** A job is added to the calendar.

***S36PRC:** A System/36 procedure is added to the calendar.

DATE

Specifies the date on which the item will occur. For recurring items, this is the date of the item's first occurrence. Up to 50 dates can be specified.

Note: The number of dates specified on this parameter must equal the number of times specified on the TIME parameter. If a list of dates is specified on this parameter, RCR(*SPECIFIC) must also be specified.

***CURRENT:** The current date is used.

date: Specify a date in the job date format.

TIME

Specifies the starting and ending times for events and meetings, or the starting time for a job or System/36 procedure. A maximum of 50 starting times and ending times can be specified.

Note: The number of times specified on this parameter must equal the number of dates specified on the DATE parameter.

If a list of times is specified on this parameter, RCR(*SPECIFIC) must also be specified.

Element 1: Start Time

start-time: Specify the start time in the 24-hour clock format.

Element 2: End Time

end-time: Specify the end time in the 24-hour clock format.

ITMID

Specifies the item identifier assigned to the item created. This parameter is valid only for creating a single item on one calendar. This parameter is used when an existing item is deleted and a new item is created with the same identifier.

***GEN:** The system generates the item identifier. The item identifier is returned as specified on the OUTFILE parameter.

item-identifier: Specify the item identifier.

Note: If the item identifier specified matches the item identifiers for a recurring set of items, the type of the item being added (as specified on the ITMTYP parameter) must match the type of the items in the recurring set.

The item identifier must be less than or equal to the highest item identifier already on the calendar.

ITMSEC

Specifies the security level of the item added.

***UNCLASSIFIED:** The item is unclassified.

***CONFIDENTIAL:** The item is confidential.

***PERSONAL:** The item is personal.

RCR

Specifies whether the item added occurs repeatedly.

***NONE:** The item added does not recur.

***DAILY:** The item occurs daily.

***WEEKLY:** The item occurs weekly.

***BIWEEKLY:** The item occurs every other week.

***MONTHLY:** The item occurs on the same date every month.

***MONTHLYREL:** The item occurs on the same relative day every month, such as the first Monday.

ADDCALITM

***SPECIFIC:** The item occurs on the dates and times specified on the DATE and TIME parameters.

ENDDATE

Specifies the last date on which a recurring item occurs. The date must be specified in the job date format. This parameter is valid only if a value other than *NONE or *SPECIFIC is specified on the RCR parameter. The item is added to the calendar on all of the dates from the date specified on the DATE parameter through the date specified on this parameter, based on the value specified on the RCR parameter, up to a maximum of 500 occurrences.

SKIPDAY

Specifies the days to skip for an item that occurs every day.

Note: This parameter is valid only if *DAILY is specified on the RCR parameter. A maximum of five values, other than *NONE, can be specified.

***NONE:** No days are skipped.

***SUN:** Sundays are skipped.

***MON:** Mondays are skipped.

***TUE:** Tuesdays are skipped.

***WED:** Wednesdays are skipped.

***THU:** Thursdays are skipped.

***FRI:** Fridays are skipped.

***SAT:** Saturdays are skipped.

MONTHWK

Specifies whether an item that occurs repeatedly on the same relative day of the month is scheduled for the fourth week or last week of the month.

Note: This parameter is valid only if *MONTHLYREL is specified on the RCR parameter and if a date later than the 25th of the month is specified on the DATE parameter.

4: The item occurs on the same relative day in the fourth week of the month.

5: The item occurs on the same relative day in the last week of the month.

TEXT

| Specifies the text that describes the item
| being added.

Note: *TEXTLIST must be specified on this parameter if text is entered in 50-character lines on the TEXTLIST parameter.

If *MEETING is specified on the ITMTYP parameter, the text entered is the purpose of the meeting.

***BLANK:** No text is used.

***TEXTLIST:** The text entered in the 50-character lines on the TEXTLIST parameter is used.

'description': Specify the text. A maximum of 3000 characters can be specified.

TEXTLIST

Specifies the text that describes the item being added. Text can be entered in 50-character lines.

The TEXT parameter allows the user to enter the text in one long series of characters, which the system separates into 50-character lines. A maximum of 100 lines (98 lines if the item is a meeting) with a maximum of 50 characters on each line can be specified.

Note: This parameter is valid only when *TEXTLIST is specified on the TEXT parameter.

If *MEETING is specified on the ITMTYP parameter, the text entered is the purpose of the meeting. Up to 98 lines can be entered.

***BLANK:** A blank line is used.

'description': Specify the text.

RMDMSG

Specifies whether a reminder message is sent to the owner of the calendar before a calendar item is scheduled to occur.

Note: This parameter is valid only if *EVENT, *MEETING, or *REMINDER is specified on the ITMTYP parameter.

***CALENDAR:** The value specified on the SNDMSG parameter of the Create Calendar (CRTCAL) command is used. (This value may have been changed since the calendar was created; the current value is used.)

***YES:** A reminder message is sent.

***NO:** No reminder message is sent.

RMDMSGTIME

Specifies the date and time the reminder message is sent.

Note: This parameter is valid only if *CALENDAR or *YES is specified on the RMDMSG parameter. This parameter is ignored if *CALENDAR is specified on the RMDMSG parameter and if the calendar referred to specifies that no reminder message is sent.

***CALENDAR:** The reminder message is sent at the time specified on the RMDLDTIME parameter of the Create Calendar (CRTCAL) command. If this value is changed since the calendar was created, the current value is used.

Element 1: Date

***ITEMDATE:** The reminder message is sent the day the item is scheduled to occur.

date: Specify the date on which the reminder message is sent. The date must be specified in the job date format.

Element 2: Time

time: Specify the time the reminder message is sent.

ITMSTS

Specifies the status of the item being added.

Note: This parameter is valid only if *EVENT or *MEETING is specified on the ITMTYP parameter.

***TENTATIVE:** The date and time of the item may change.

***CONFIRMED:** The date and time of the item will not change.

MTGSUB

Specifies the subject of the meeting.

Note: This parameter is valid only if *MEETING is specified on the ITMTYP parameter.

***BLANK:** The subject is left blank.

'meeting-subject': Specify a meeting subject of up to 50 characters.

MTGPLC

Specifies where the meeting will take place.

Note: This parameter is valid only if *MEETING is specified on the ITMTYP parameter.

***BLANK:** The place is left blank.

'meeting-place': Specify the meeting place. Up to 50 characters can be specified.

MTGREQ

Specifies who is requesting the meeting.

Note: This parameter is valid only if *MEETING is specified on the ITMTYP parameter.

***CURRENT:** The user issuing the command is requesting the meeting.

Element 1: User ID

user-ID: Specify the user ID of the person requesting the meeting.

Element 2: User Address

user-address: Specify the user address of the person requesting the meeting.

JOB

Specifies the name and library of the job description of the job that is scheduled to be run.

Note: This parameter is valid only if *JOB is specified on the ITMTYP parameter.

The possible library values are:

***LIBL:** The library list is used to locate the job description.

***CURLIB:** The current library for the job is used to locate the job description. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the job description is located.

QDFTJOB: The system-supplied job description, QDFTJOB, is used.

job-description-name: Specify the name of the job description.

ADDCALITM

JOBCMD

Specifies the name of the CL command to be run in the scheduled job. Up to 256 characters can be specified.

Note: This parameter is valid only if *JOB is specified on the ITMTYP parameter.

S36PRC

Specifies the name and library of the System/36 procedure to be run.

Note: This parameter is valid only if *S36PRC is specified on the ITMTYP parameter.

The possible library values are:

***LIBL:** The library list is used to locate the procedure.

***CURLIB:** The current library for the job is used to locate the procedure. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the procedure is located.

System/36-procedure-name: Specify the name of the System/36 procedure.

S36PRCPTY

Specifies the priority at which the System/36 procedure is run. The highest priority is 5; the lowest priority is 0.

Note: This parameter is valid only if *S36PRC is specified on the ITMTYP parameter.

3: The priority is 3.

0: The priority is 0.

1: The priority is 1.

2: The priority is 2.

4: The priority is 4.

5: The priority is 5.

S36PRCPRM

Specifies the parameters that are passed to the System/36 procedure when it is run.

Note: This parameter is valid only if *S36PRC is specified on the ITMTYP parameter.

***NONE:** No parameters are passed to the procedure.

procedure-parameters: Specify the parameters that are passed to the procedure. Up to 256 characters can be specified.

OUTFILE

Specifies the name and library of the database file to which the output of this command is directed. The database file contains the item identifiers and the names of the calendars to which items are added. If the file does not exist, the system creates a file in the specified library. If a new file is created, the system uses QAOYADDITM in QOFC with a record format name of ADDITM as a model. If the file already exists, it must have this format.

***NONE:** The output is not directed to a database file. A message is sent that lists the number of items added.

The possible library values are:

***LIBL:** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file is created in the QGPL library.

***CURLIB:** The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the file is located.

database-file-name: Specify the name of the output file.

OUTMBR

Specifies the name of the database file member that receives the output of this command.

Element 1: Member to Receive Output

***FIRST:** The first member in the file receives the output. If it does not exist, the system creates a member with the same name as the file specified on the OUTFILE parameter.

member-name: Specify the name of the file member that receives the output. If it does not exist, the system creates it.

Element 2: Operation to Perform on Member

***REPLACE:** The output data replaces existing records in the specified member.

***ADD:** The output data is added at the end of existing records in the specified member.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEVd:** The character set and code page defined for the device are used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples

Example 1: Adding An Item

```
ADDCALITM CAL(*SCHED) DATE(022790)
TIME(('8:00' '10:00')) ITMTYP(*EVENT)
SNDMSG(*NO) ITMSTS(*CONFIRMED)
TEXT('V2R1 Calendar API I0')
```

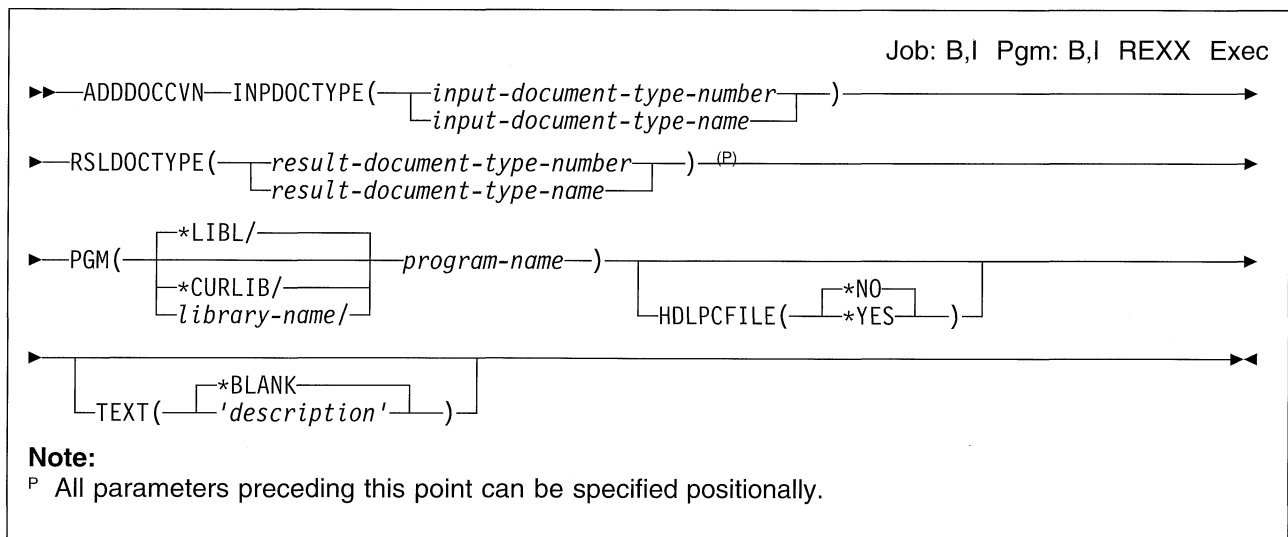
This command adds an item to the scheduling calendar of the user issuing the command. The event will be confirmed from 8:00 a.m. to 10:00 a.m. on February 27, 1990. The text for the event is 'V2R1 Calendar I0'.

Example 2: Adding a Meeting

```
ADDCALITM CAL(*GRP) GRPCALS(DEPT54P)
DATE(010290) TIME(('13:00' '14:00'))
RCR(*WEEKLY) ENDDATE(123190)
ITMTYP(*MEETING) MTGSUB('Dept mtg')
MTGPLC('Bob's office')
TEXT('Discuss scheduling')
```

This command adds a meeting to the calendar listed in the calendar group, DEPT54P. The meeting will be a weekly meeting on Tuesday (January 2, 1990 is a Tuesday) from 1:00 p.m. to 2:00 p.m. The subject of the meeting is 'Dept mtg' and it is held in Bob's office. The purpose of the meeting is to 'discuss scheduling'.

ADDDOCCVN (Add Document Conversion) Command



Purpose

The Add Document Conversion (ADDDOCCVN) command adds a document conversion definition to the system and defines the attributes of the document conversion.

Restriction: You must have security officer (*SECOFR) authority or be enrolled in OfficeVision/400 as an administrator to use this command.

Parameters

INPDOCTYPE

Specifies the name or number of the document type used as input for the document conversion definition being added.

input-document-type-number: Specify the numeric value associated with the Document Interchange Architecture (DIA) document type requested. Valid values range from 1 through 65535.

input-document-type-name: Specify the 10-character name associated with the defined document type.

RSLDOCTYPE

Specifies the name or number of the document type used as the result type for the document conversion being added.

result-document-type-number: Specify the numeric value associated with the DIA document type requested. Valid values range from 1 through 65535.

result-document-type-name: Specify the 10-character name associated with the defined document type.

PGM

Specifies the user document conversion program to be called when OfficeVision/400 needs to convert a document.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the program is located.

program-name: Specify the name of the conversion program.

Parameters

HDLPCFILE

Specifies the whether the document conversion is able to accept PCFILE documents as an alternative form of input document type.

This allows multiple document conversions to be defined to handle the PCFILE documents.

***NO:** This conversion does not support PCFILE as an alternative document type.

***YES:** This conversion accepts PCFILE documents and the document types specified on the INPDOCTYPE parameter as input.

TEXT

| Specifies the text that briefly describes the
| document conversion. This parameter can be
| DBCS enabled.

***BLANK:** No text is specified.

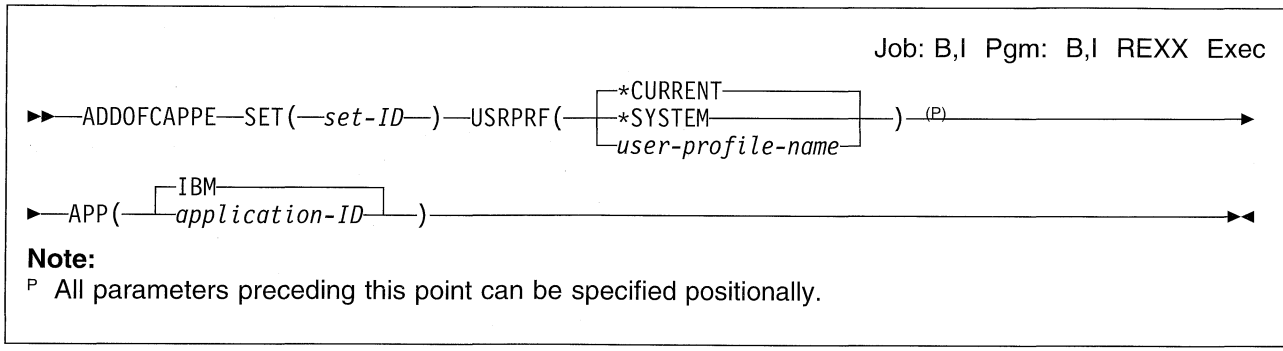
'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
ADDDOCCVN INPDOCTYPE(32775) RSLDOCTYPE(RFTDCA)  
PGM(WPX/XF1)
```

This command adds a document conversion from document type 32775 to document type RFTDCA. The program XF1 in library WPX performs the operation.

ADDOFCAPPE (Add Office Application Entry) Command



Purpose

The Add Office Application Entry (ADDOFCAPPE) command allows the user to associate applications and sets for a specified user.

Parameters

SET

Specifies the set associated with the user.

set-ID: The name of an existing set to be associated with an application.

USRPRF

Specifies the user for whom the office application entry is to be added.

***CURRENT**: The application entries for the current signed on user are added.

Note: If the user is not the security officer nor does the user have *SECADM authority, *CURRENT must be specified for this parameter.

***SYSTEM**: The system default values are used if the user has not requested an application for a set.

user-profile-name: Specify the name of the user profile used.

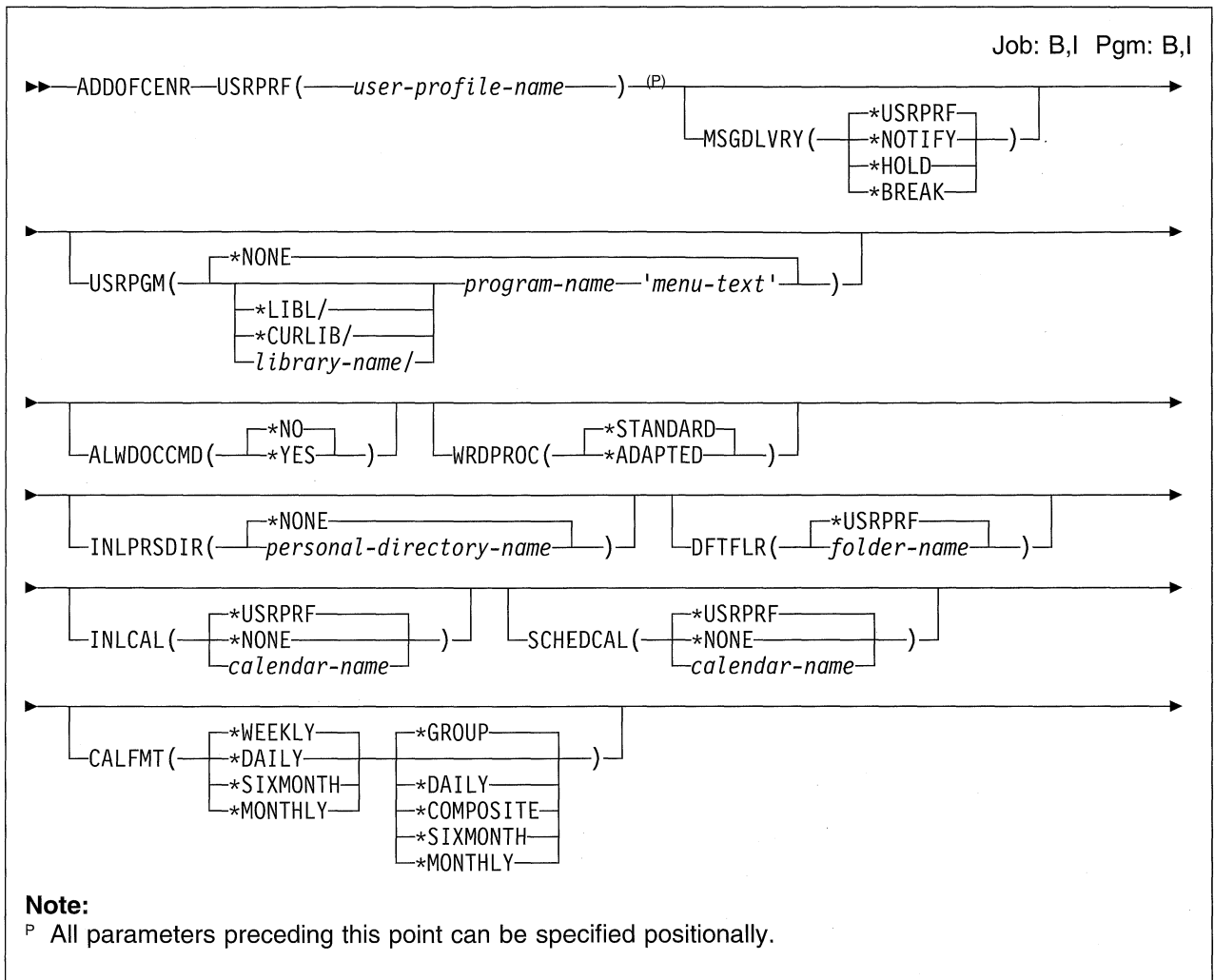
APP

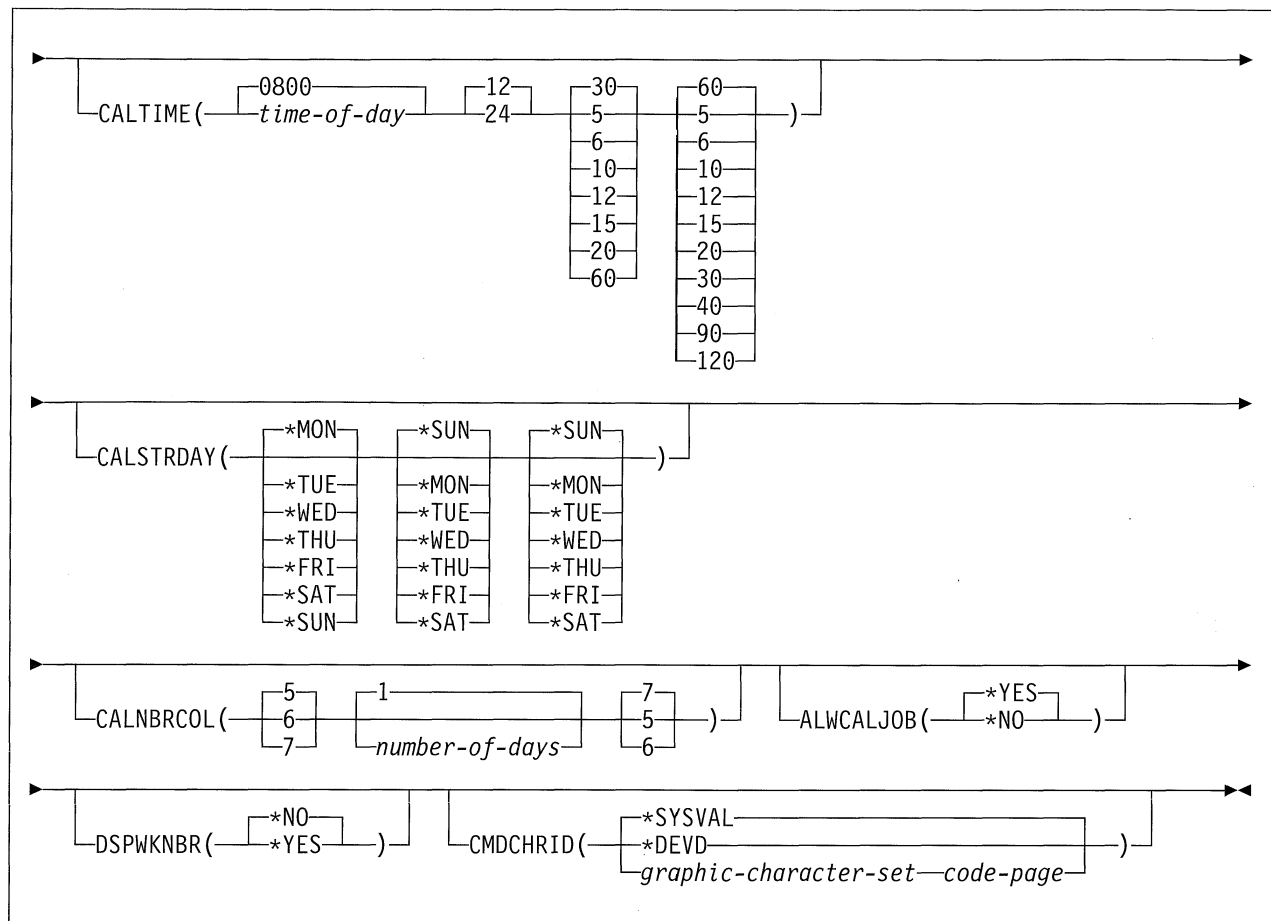
Specifies the identifier of the application being added. The application name must conform to system object naming conventions.

IBM: The OfficeVision/400 editor support is associated to this set.

application-ID: Specify the ID of the application used.

ADDOFCENR (Add Office Enrollment) Command





Purpose

The Add Office Enrollment (ADDOFCENR) command enrolls a user in OfficeVision/400. The user being enrolled must already have a user profile and a System Distribution Directory entry. Ignore the CAL and MAIL parameters when these options are not installed on the system.

Restriction: The user must have *SECADM authority and be enrolled in OfficeVision/400 as an administrator.

Parameters

USRPRF

Specifies the name of the user profile of the user being enrolled in OfficeVision/400.

MSGDLVRY

Specifies how a user is notified of messages while using OfficeVision/400.

***USRPRF:** The delivery mode is the same as the mode specified in the user profile.

***NOTIFY:** The audible alarm is sounded or the message waiting light is turned on. The light or alarm is turned off when the user looks at the message.

***HOLD:** Messages are held in the message queue until they are requested by the user or program.

***BREAK:** The user's job is interrupted and the message is displayed.

USRPGM

Specifies the qualified name of the user program to be called when the user selects Option 50 on the OfficeVision/400 menu, and the text that is shown next to Option 50. No check is made to verify that the specified user program exists.

***NONE:** No user program name is specified.

Element 1: User Program Name

The possible library values are:

***LIBL:** The library list is used to locate the user program.

***CURLIB:** The current library for the job is used to locate the user program. If no library is specified as the current library for the job, QGPL is used.

library-name: Specify the name of the library where the user program is located.

program-name: Specify the name of the user program. If this value is specified, the menu text value must also be specified.

Element 2: Menu Text

'menu-text': Specify the text that is shown next to Option 50 on the OfficeVision/400 menu. The text can be a maximum of 40 characters, enclosed in apostrophes. If this value is specified, the program name value must also be specified.

ALWDOCCMD

Specifies whether the user can use control language (CL) commands when working with text documents.

***NO:** The user cannot use CL commands when working with text documents.

***YES:** The user can use CL commands when working with text documents.

WRDPROC

Specifies the type of word processing function used.

***STANDARD:** The word processing function that is appropriate for the user's work station is used. If the user works on a personal computer, the PC text-assist function is used.

***ADAPTED:** The adapted word processing function is used, regardless of the type of work station.

INLPRSDIR

Specifies the name of the first personal directory the user will use when running personal directory functions. If *NONE is specified, the user must select a personal directory each time the personal directory function is used. No check is made to verify that the specified personal directory exists.

***NONE:** No personal directory is used.

personal-directory-name: Specify the name of the personal directory.

DFTFLR

Specifies the name of the folder that is used whenever a folder is needed but not specified. If the folder does not exist, it is created.

***USRPRF:** The name of the folder is the same as the user profile name.

folder-name: Specify the name of the folder.

INLCAL

Specifies the name of the first calendar the user will see when using calendar functions. The name of a calendar, calendar group, or a distribution list calendar can be specified. If the calendar specified does not exist, it is created.

***USRPRF:** The name of the calendar is the same as the user profile name.

***NONE:** No calendar name is specified.

calendar-name: Specify the name of the calendar. The name can be a maximum of 10 characters.

SCHEDCAL

Specifies the name of the user's scheduling calendar. If the calendar specified does not exist, it is created.

***USRPRF:** The name of the calendar is the same as the user profile name.

***NONE:** No calendar name is specified.

calendar-name: Specify the name of the calendar. The name can be a maximum of 10 characters.

CALFMT

Specifies the type of view shown when only one calendar is used, and the type of view shown when more than one calendar is used.

Element 1: One-calendar View

***WEEKLY:** A weekly calendar is shown.

***DAILY:** A daily calendar is shown. A daily calendar shows a detailed view of the calendar for a single day or several days.

***SIXMONTH:** A six-month calendar is shown. A six-month calendar shows six months on one display. It can be used to check dates, count the number of days from one date to another, or for any other task that requires a monthly format.

***MONTHLY:** A monthly calendar is shown. This calendar can be used to view the calendar items for one month.

Element 2: Multiple-calendar View

***GROUP:** A group calendar is shown. A group calendar shows a single day with each of the calendars' schedules for that day displayed in a separate column.

***DAILY:** A daily calendar is shown. A daily calendar shows a detailed view of a single day with each of the calendars' schedules for that day displayed consecutively.

***COMPOSITE:** The available times on up to 10 calendars are shown.

***SIXMONTH:** A six-month calendar is shown. A six-month calendar shows six months on one display. It can be used to check dates, count the number of days from one date to another, or for any other task that requires a monthly format.

***MONTHLY:** Specifies a monthly view of the calendar items.

CALTIME

Specifies the time of day at which the calendar starts, the format in which time is shown to the user, and the number of minutes in each time slot on the calendar.

Element 1: Calendar Start Time

0800: The calendar starts at 8 a.m.

time-of-day: Specify the time of day at which the calendar starts.

Element 2: Calendar Time Format

12: A 12-hour day is used.

24: A 24-hour day is used.

Element 3: Calendar Time Interval

30: Each time slot is 30 minutes long.

number-of-minutes: Specify the number of minutes in each time slot. Valid values are: 5, 6, 10, 12, 15, 20, 30, and 60.

Element 4: Monthly View Time Interval

60: Each time slot is 60 minutes long.

number-of-minutes: Specify the number of minutes in each time slot for the monthly calendar view. The valid values are 5, 6, 10, 12, 15, 20, 30, 40, 60, 90, and 120.

CALSTRDAY

Specifies the day of the week shown in the left column of the weekly calendar, and the day of the week shown in the left column of the six-month calendar.

Element 1: Weekly View Calendar Start Day

***MON:** Monday is shown in the left column of the calendar.

***TUE:** Tuesday is shown in the left column of the calendar.

***WED:** Wednesday is shown in the left column of the calendar.

***THU:** Thursday is shown in the left column of the calendar.

***FRI:** Friday is shown in the left column of the calendar.

***SAT:** Saturday is shown in the left column of the calendar.

***SUN:** Sunday is shown in the left column of the calendar.

Element 2: Six-month Calendar Start Day

***SUN:** Sunday is shown in the left column of the calendar.

***MON:** Monday is shown in the left column of the calendar.

***TUE:** Tuesday is shown in the left column of the calendar.

***WED:** Wednesday is shown in the left column of the calendar.

***THU:** Thursday is shown in the left column of the calendar.

***FRI:** Friday is shown in the left column of the calendar.

***SAT:** Saturday is shown in the left column of the calendar.

Element 3: Monthly View Start Day

***SUN:** Sunday is shown in the left column of the calendar.

***MON:** Monday is shown in the left column of the calendar.

***TUE:** Tuesday is shown in the left column of the calendar.

***WED:** Wednesday is shown in the left column of the calendar.

***THU:** Thursday is shown in the left column of the calendar.

***FRI:** Friday is shown in the left column of the calendar.

***SAT:** Saturday is shown in the left column of the calendar.

CALNBRCOL

Specifies the number of columns that are shown on the calendar for weekly, group, or composite views, and the number of days shown on the daily view. Depending on the value specified on the CALFMT parameter, each column represents one of the following:

- A consecutive day for a weekly calendar
- A different user for the same day for a group calendar
- A consecutive day containing calendar items for up to 10 users for a composite calendar

Element 1: Number of Columns

5: Five columns are shown.

6: Six columns are shown.

7: Seven columns are shown.

Element 2: Number of Days

1: One day is shown on the daily calendar.

number-of-days: Specify the number of days shown on the daily calendar. Valid values range from 1 through 999.

Element 3: Number of Monthly View Columns

7: Seven columns are shown.

5: Five columns are shown.

6: Six columns are shown.

ALWCALJOB

Specifies whether the user can schedule jobs and procedures on calendars.

***YES:** The user can schedule jobs and procedures.

***NO:** The user cannot schedule jobs and procedures.

DSPWK NBR

Specifies whether to display the week number.

***NO:** Do not display the week number.

***YES:** Display the week number.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being entered as command parameter values. This character identifier is related to the display device used to specify the command. More information about CHRID processing is in the *Guide to Programming Displays*.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device are used.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

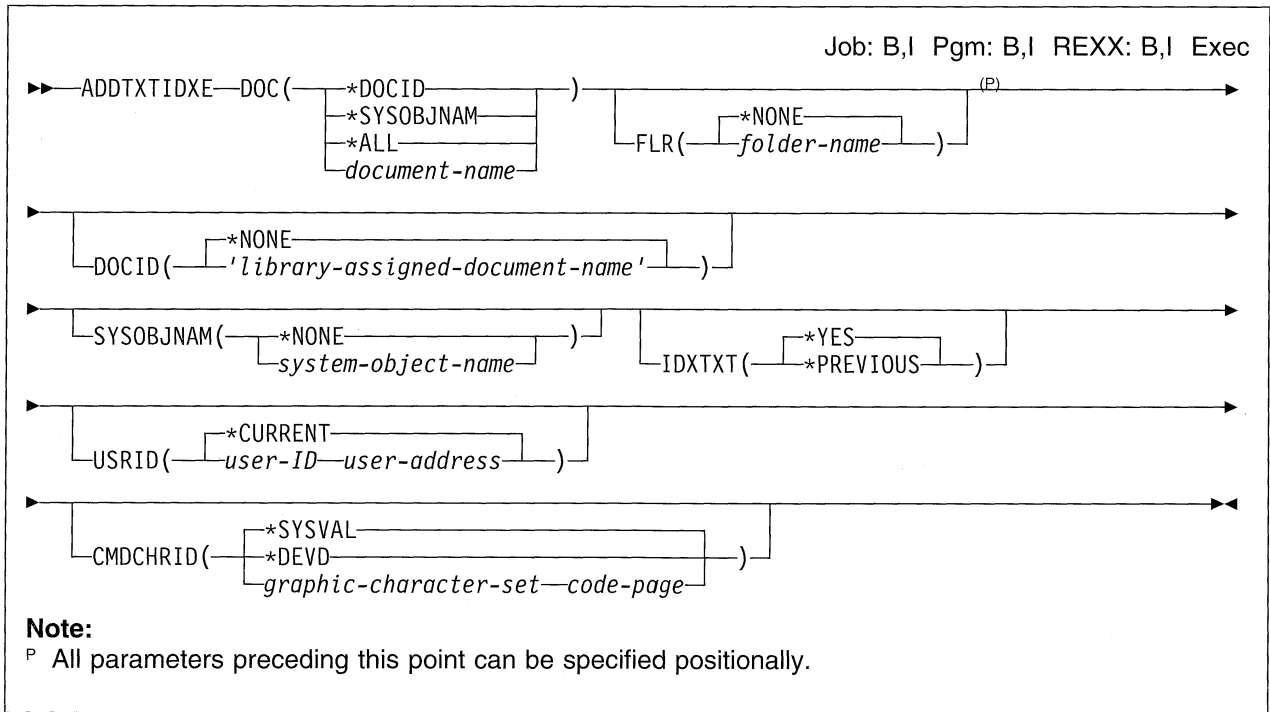
code-page: Specify the code page. Valid values range from 1 through 9999.

Example

```
ADDOFCENR USRPRF(JACKSON)
WRDPROC(*STANDARD) DFTFLR (JACKSON)
INLCAL(JACKSON) SCHEDCAL(JACKSON)
CALFMT(*WEEKLY) CALTIME('08:00' 12 30)
CALSTRDAY(*MON) CALNBRCOL(5 1)
```

This command enrolls user Jackson in OfficeVision/400 with standard word processing functions; a default folder, calendar, and personal calendar named Jackson; a Monday-through-Friday calendar, displayed one day at a time, that starts each day at 8 a.m. and has a slot for each 30-minute period in the 12-hour day.

ADDTXTIDX (Add Text Index Entry) Command



Purpose

The Add Text Index Entry (ADDTXTIDX) command places indexing requests on a scheduling queue. Documents are indexed during the next scheduled index update. Use the Display Index Status (DSPIDXSTS) command to display the update schedule. If a document is indexed, it can be found by:

- Using the Query Document Library (QRYDOCLIB) command. The document is eligible to be selected if it meets the criteria specified on the QRYTXT parameter.
- Specifying document text criteria when searching for documents with OfficeVision/400.

The document can be identified by:

- The document name (DOC) and the folder (FLR) in which the document exists
- The library-assigned document name using DOC(*DOCID) and the DOCID parameter
- The system object name using DOC(*SYSOBJNAM) and the SYSOBJNAM parameter

Restriction: The user, or the user for whom the request is being made, must have at least *USE authority for the document or *ALLOBJ special authority.

Parameters

DOC

Specifies the name of the document to be indexed.

***DOCID:** The document specified on the DOCID parameter is indexed.

***SYSOBJNAM:** The document specified on the SYSOBJNAM parameter is indexed.

***ALL:** All documents in the folder specified on the FLR parameter are indexed.

document-name: Specify up to 12 characters for the name of the document to be indexed.

FLR

Specifies the name of the folder that contains the documents to be indexed. A folder name must be specified if a user-assigned document or *ALL is specified on the DOC parameter.

***NONE:** This value is specified if *DOCID or *SYSOBJNAM is specified on the DOC parameter.

folder-name: Specify a maximum of 63 characters for the user-assigned name of the folder containing the document. If the document is nested within folders, all of the associated folders must be specified, in the form FLR1/FLR2/FLRn, where FLRn is the folder in which the document exists.

DOCID

Specifies the library-assigned name of the document to be indexed. A library-assigned document name must be specified if *DOCID is specified on the DOC parameter.

A library-assigned document name is a system-generated name based on the date and time that the document was created. Library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format: YYYYMMDDHHMNSSHSSNSNSNSN where :

- YYYY = year
- MM = month
- DD = day
- HH = hour
- MN = minute
- SS = second
- HS = hundredths of a second
- SNSNSNSN = system name

***NONE:** No library-assigned document name is specified.

'library-assigned-document-name': Specify the library-assigned name of the document, enclosed in apostrophes.

SYSOBJNAM

Specifies the system object name of the document to be indexed. A system object name must be specified if *SYSOBJNAM is specified on the DOC parameter.

***NONE:** No system object name is specified.

system-object-name: Specify the system object name of the document.

IDXTXT

Specifies whether text index entries are created for all documents or just those that had previously been indexed.

***YES:** All documents are indexed.

***PREVIOUS:** Documents that were previously indexed are re-indexed. Documents that were not previously indexed are not indexed.

USRID

Specifies the user ID and user address associated with the request.

Note: To work on another's behalf you must have been given permission with the Grant User Permission (GRTUSRPMN) command.

***CURRENT:** You are making the request for yourself.

Element 1: User ID

user-ID: Specify your user ID or the user ID of the user for whom you are making the request.

Element 2: User Address

user-address: Specify your user address or the user address of the user for whom you are making the request. *user-address:* Specify another's user address or your user address.

CMDCHRID

Specifies the graphic character set and code page of the characters entered as parameter values. The character set and code page are related to the type of display device used.

***SYSVAL:** The graphic character set and code page are taken from the QCHRID system value.

***DEV:** The graphic character set and code page are taken from the display device description. This value is valid only in the interactive environment.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 999 characters.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 999 characters.

ADDTXTIDX

Examples

Example 1: Indexing a Document in a Folder

```
ADDTXTIDX DOC(ABC) FLR(XYZ)
```

This command indexes document ABC, which exists in folder XYZ.

Example 2: Indexing all Documents in a Folder

```
ADDTXTIDX DOC(*ALL) FLR(XYZ)
```

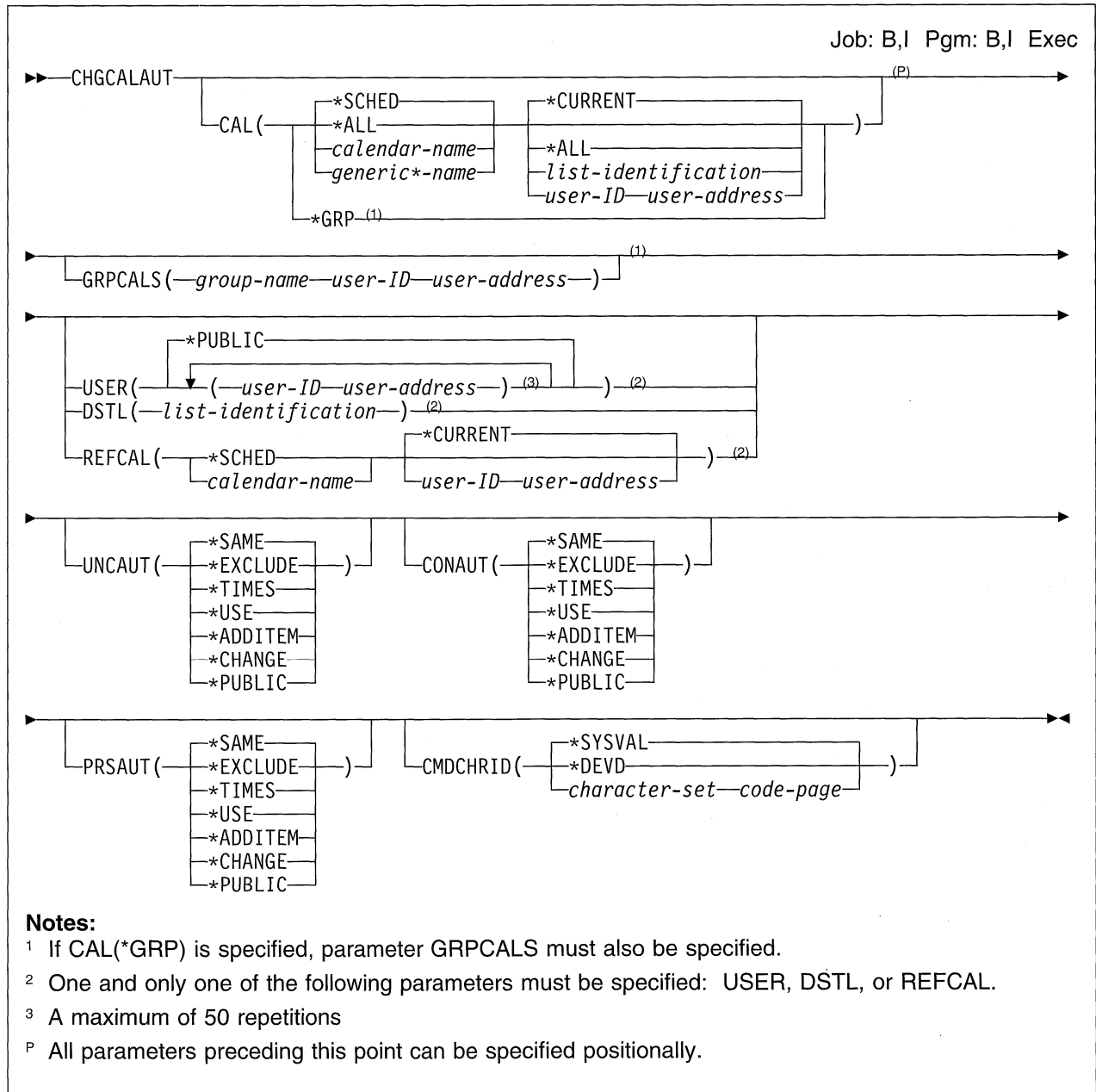
This command indexes all documents in folder XYZ.

Example 3: Re-indexing all Documents in a Folder

```
ADDTXTIDX DOC(*ALL) FLR(XYZ) IDXTXT(*PREVIOUS)
```

This command indexes all of documents in folder XYZ that were previously indexed. The documents in XYZ that were not previously indexed are not indexed.

CHGCALAUT (Change Calendar Authority) Command



Purpose

The Change Calendar Authority (CHGCALAUT) command allows the user to change the public and private authorities to one or more calendars. The calendar authorities can be changed by distribution list, list of users, or a reference calendar.

Restrictions: The user must be the owner of the calendar, have *SECADM authority, or have *ALLOBJ authority to use this command.

The user must specify a value for one of the following parameters: USER, DSTL, or REFCAL.

Parameters

CAL

Specifies the name and owner of the calendar for which the authorities are to be changed. A calendar name and owner name, or *GRP, can be specified on this parameter.

Element 1: Calendar Name

***SCHED:** The authorities for the scheduling calendars owned by the users specified in Element 2 of this parameter are changed.

***ALL:** The authorities for all calendars owned by the users specified on Element 2 of this parameter are changed.

calendar-name: Specify the name of the calendar for which the authorities are changed.

generic-name:* Specify the generic name of the calendar. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all calendars with names that begin with the generic name, and for which the user has authority, are changed. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete calendar name.

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify 'MIKE**' to change authorities for all calendars that have names beginning with 'MIKE*'.

Element 2: Calendar Owner

***CURRENT:** The authorities for the calendars owned by the user issuing the commands.

***ALL:** The authorities for the calendars that have the name specified on Element 1 of this parameter and are owned by local users are changed.

Note: The user must have *SECADM or *ALLOBJ special authority to specify *ALL for both the calendar name and the owner.

list-identification: Specify a distribution list. The authorities for calendars specified on Element 1 of this parameter and owned by users on the distribution list are changed.

user-ID: Specify the user ID of the owner of the calendars to be changed. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendars to be changed. This value is valid only if user ID is specified.

Other Single Values

***GRP:** The authorities for the calendars in the calendar group specified on the GRPCALS parameter are changed.

GRPCALS

Specifies the local calendar group for which authority is being changed. This parameter is valid only if CAL(*GRP) is also specified. A value for Element 1 must be specified if GRPCALS is specified.

Element 1: Group Name

group-name: Specify the local calendar group for which authorities are being changed.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group for which authorities are being changed.

Element 3: Address

user-address: Specify the user address of the owner of the calendar group for which authorities are being changed.

USER

Specifies a list of users whose authorities are being changed.

Note: If USER is specified, the DSTL and REFCAL parameters cannot be specified.

***PUBLIC:** The public authority to the calendars is changed.

user-ID: Specify the user IDs of the users whose authority is being changed. This value is valid only if user address is also specified. A maximum of 50 users can be specified.

user-address: Specify the user addresses of the users whose authority is being changed. This value is valid only if user ID is also specified. A maximum of 50 users can be specified.

DSTL

Specifies the name of the distribution list that contains the list of users whose authorities are being changed.

Note: If DSTL is specified, the USER and REFCAL parameters cannot be specified.

REFCAL

Specifies the name of a reference calendar from which the private authorities are copied.

Note: The public authority to the calendars specified on the CAL parameter of this command is not changed by this parameter.

If a user has already been assigned a private authority to the calendars specified on the CAL parameter of this command, the authority is not changed unless the user also has been assigned a private authority to the reference calendar. In that case, the user's private authority to the calendars specified on the CAL parameter of this command is replaced by the authority from the reference calendar.

If REFCAL is specified, the USER, DSTL, UNCAUT, CONAUT, and PRSAUT parameters cannot be specified.

Element 1: Calendar Name

***SCHED:** The private authorities are copied from the scheduling calendar owned by the user specified in Element 2 of this parameter.

calendar-name: Specify the name of the reference calendar.

Element 2: Calendar Owner

***CURRENT:** The private authorities are copied from the calendar owned by the user specified on Element 1 of this parameter.

user-ID: Specify the user ID of the owner of the reference calendar. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the reference calendar. This value is valid only if user ID is also specified.

UNCAUT

Specifies the authority to unclassified items given to users specified on the DSTL parameter or on the USER parameter.

Note: If REFCAL is specified, the UNCAUT, CONAUT, and PRSAUT parameters cannot be specified.

***SAME:** The authority does not change.

***EXCLUDE:** The user cannot see unclassified calendar items.

***TIMES:** The user can see the times for which unclassified items on the calendar are scheduled, but cannot see any text associated with the items.

***USE:** The user can see the times for which unclassified items on the calendar are scheduled and the text associated with the items.

***ADDITEM:** The user is given *USE authority to the calendar. The user also can add items to the calendar and change the items added.

***CHANGE:** The user can change or delete unclassified calendar items. The user cannot delete the calendar itself or change the list of users authorized to the calendar.

***PUBLIC:** The authority to unclassified calendar items for the specified users is changed to the same authority as the public authority.

CONAUT

Specifies the authority for confidential items given to users specified on the DSTL parameter or on the USER parameter.

Note: If REFCAL is specified, the UNCAUT, CONAUT, and PRSAUT parameters cannot be specified.

***SAME:** The authority does not change.

***EXCLUDE:** The user cannot see confidential calendar items.

***TIMES:** The user can see the times for which the confidential items on the calendar are scheduled, but cannot see any text associated with the items.

***USE:** The user can see the times for which confidential items on the calendar are scheduled and the text associated with the items.

***ADDITEM:** The user is given *USE authority to the calendar. The user also can add items to the calendar and change the items added.

CHGCALAUT

***CHANGE:** The user can change or delete confidential calendar items. The user cannot delete the calendar itself or change the list of users authorized to the calendar.

***PUBLIC:** The authority to confidential calendar items for the specified users is changed to the same authority as the public authority.

PRSAUT

Specifies the authority for personal items given to users specified on the DSTL parameter or on the USER parameter.

Note: If REFCAL is specified, the UNCAUT, CONAUT, and PRSAUT parameters cannot be specified.

***SAME:** The authority does not change.

***EXCLUDE:** The user cannot see personal calendar items.

***TIMES:** The user can see the times for which personal items on the calendar are scheduled, but cannot see any text associated with the items.

***USE:** The user can see the times for which personal items on the calendar are scheduled and the text associated with the items.

***ADDITEM:** The user is given *USE authority to the calendar. The user also can add items to the calendar and change the items added.

***CHANGE:** The user can change or delete personal calendar items. The user cannot delete the calendar itself or change the list of users authorized to the calendar.

***PUBLIC:** The authority to personal calendar items for the specified users is changed to the same authority as the public authority.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device is used. This value

cannot be specified if the command is run in batch.

Element 1: Character Set

character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples

Example 1: Changing Public Authorities for Scheduling Calendar

```
CHGCALAUT CAL(*SCHED) USER(*PUBLIC)
PRSAUT(*EXCLUDE)
```

This command changes the public authority to *EXCLUDE for the scheduling calendar of the user issuing the command. The confidential and unclassified authorities are not changed.

Example 2: Changing Authorities for Calendars in a Group

```
CHGCALAUT CAL(*GRP) GRPCALS(DEPT54P)
DSTL(DEPT54P DSTL) UNCAUT(*CHANGE)
```

This command changes the unclassified authority for the calendars listed in the calendar group DEPT54P to *CHANGE for all users listed in the distribution list DEPT54P DSTL. The personal and confidential authorities are changed to *TIMES if they are currently *EXCLUDE (to allow for a valid free time search), otherwise they are not changed.

Example 3: Changing Authorities

```
CHGCALAUT CAL(*ALL PJC RCHAS1)
REFCAL(*SCHED PJC RCHAS1)
```

This command changes all of the calendars owned by PJC to have the same authorities as the scheduling calendar for PJC. Any private authorities that were already granted to the calendars that are not in the scheduling calendar are not changed. All private authorities granted to the calendars in the scheduling calendar are replaced with the authorities in the specified scheduling calendar.

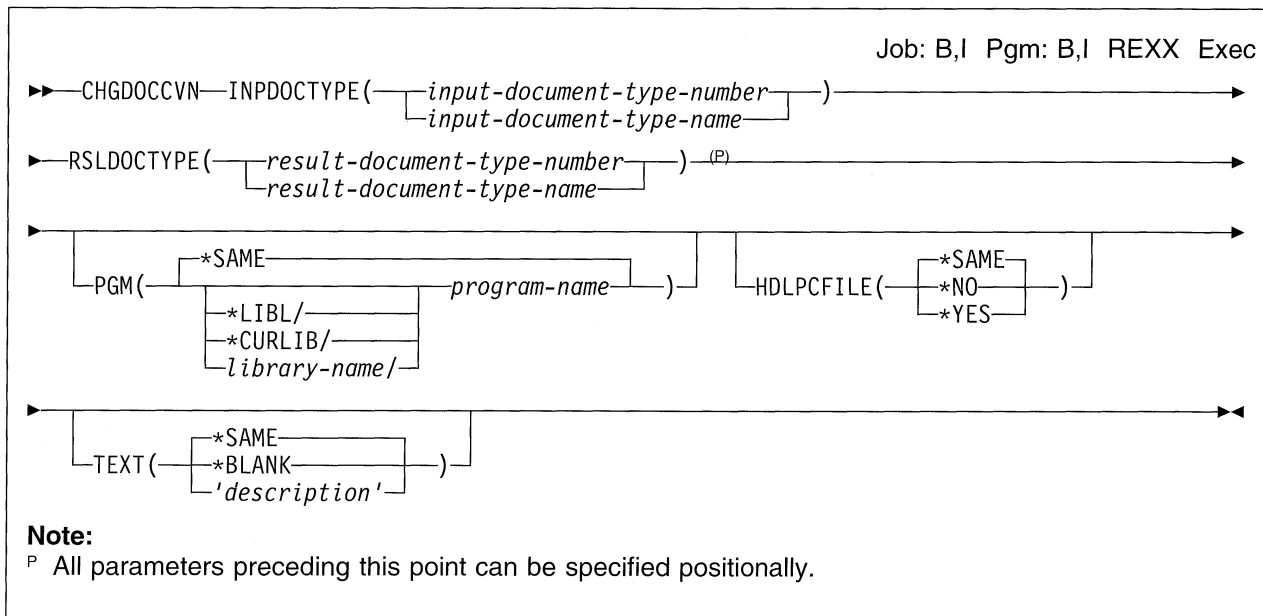
Example 4: Changing Authorities For More Than One User


```
CHGCALAUT CAL(*SCHED) USER((MITCH RCHAS1)
(PJC RCHAS1)) CONAUT(*ADDITEM)
PRSAUT(*ADDITEM)
```

This command changes the confidential and personal authorities to the scheduling calendar for the

user issuing the command for local users PJC and MITCH to *ADDITEM. The unclassified authority is changed to *ADDITEM if it is not already at least *ADDITEM.

CHGDOCCVN (Change Document Conversion) Command



Purpose

The Change Document Conversion (CHGDOCCVN) command is used to change the attributes of a document conversion definition.

Restriction: You must have security officer (*SECOFR) authority or be enrolled in OfficeVision/400 as an administrator to use this command.

Parameters

INPDOCTYPE

Specifies the name or number of the document type used as input for the document conversion definition being changed.

input-document-type-number: Specify the numeric value associated with the Document Interchange Architecture (DIA) document type requested. Valid values range from 1 through 65535.

input-document-type-name: Specify the 10-character name associated with the defined document type.

RSLDOCTYPE

Specifies the name or number of the document type used as the result type for the document conversion definition being changed.

result-document-type-number: Specify the numeric value associated with the DIA document type requested. Valid values range from 1 through 65535.

result-document-type-name: Specify the 10-character name associated with the defined document type.

PGM

Specifies the user document conversion program to be called when OfficeVision/400 needs to convert a document.

***SAME:** The value does not change.

The possible library values are:

***LIBL:** The library list is used to locate the program.

***CURLIB:** The current library for the job is used to locate the program. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the program is located.

program-name: Specify the name of the conversion program.

HDLPCFILE

Specifies the whether the document conversion is able to accept PCFILE documents as an alternative form of input document type. This allows multiple document conversions to be defined to handle the PCFILE documents.

***SAME:** The value does not change.

***NO:** This conversion does not support PCFILE as an alternative document type.

***YES:** This conversion accepts PCFILE documents and the document types specified on the INPDOCTYPE parameter as input.

TEXT

Specifies the text that briefly describes the document conversion. This parameter is not DBCS-enabled.

***SAME:** The value does not change.

***BLANK:** No text is specified.

'description': Specify no more than 50 characters of text, enclosed in apostrophes.

Example

```
CHGDOCCVN INPDOCTYPE(RFTDCA) RSLDOCTYPE(32775)
PGM(WPX/XF2)
```

This command changes the document conversion program for document type RFTDCA to XF2 in library WPX.

CHGDOCSET (Change Document Set) Command

Job: B,I Pgm: B,I REXX: B,I Exec

▶ CHGDOCSET SET (—*set-ID*—) ^(P) TEXT (

| |
|---------------|
| *SAME |
| *SET |
| *BLANK |
| 'description' |

) ▶▶

Note:
^P All parameters preceding this point can be specified positionally.

Purpose

The Change Document Set (CHGDOCSET) command allows the user to change the description of a set definition that already exists on the system.

Parameters

SET

The name of the set being changed. The name must conform to system naming conventions.

set-ID: The name of the set to change.

TEXT

The text description to be associated with the set.

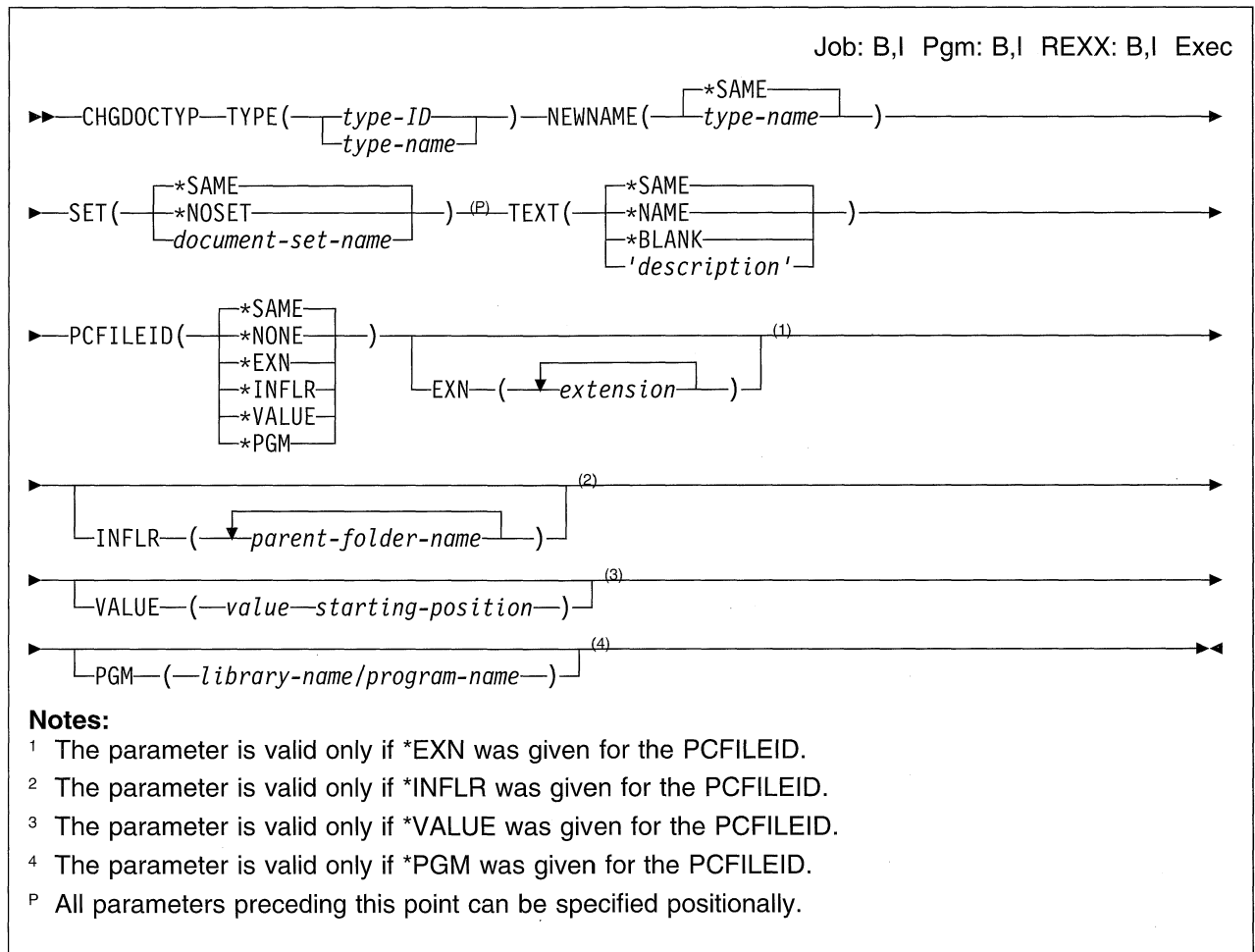
***SAME:** Do not change the text associated with the set.

***SET:** The text will be identical to the set name.

***BLANK:** No text will be specified.

'description': The description to be assigned. It will be kept in the input CCSID and converted if presented by a job supporting a different CCSID.

CHGDOCTYP (Change Document Type) Command



Purpose

The Change Document Type (CHGDOCTYP) command allows the user to change an existing type definition.

Parameters

TYPE

Specifies the name or number of the document type being changed.

type-ID: The numeric value between 1 and 65535 that is associated with the DIA document type being changed.

type-name: The name of the type being changed.

NEWNAME

Specifies the new name to be associated with the type.

***SAME**: The name of the type is not changed.

type-name: The alphanumeric name to be associated with this document type. The name is 1 to 10 characters long, it must meet the system object naming conventions, and it must be unique.

Note: The names of IBM-supplied document types will be translated for each supported language. Because of this, the uniqueness of a new name is assured only with other user assigned names within the currently used language. It is possible to assign a name that will conflict with that of another language.

CHGDOCTYP

When another language is in use, the name will refer to the IBM-supplied type.

SET

Specifies the set to which the type belongs.

***SAME:** The set membership of the type is not changed.

***NOSET:** This type belongs to no set.

document-set-name: The name of an existing set to which this type belongs.

TEXT

Specifies the text description assigned to this type. If the screen has enough space, this text will be shown and will identify the type.

***SAME:** The text associated with the type will not be changed.

***NAME:** The text will be identical to the name assigned to the type.

***BLANK:** No text will be specified.

'description': The description will be assigned. It will be kept in the input CCSID and converted if ever presented by a job supporting a different CCSID.

PCFILEID

Specifies the identification values for changing PCFILE type documents to be the document type specified on TYPE. The values used to make the determination are the document name extension, the parent folder name, a data string within the document, or the results from calling an external program.

***SAME:** The current value will not change.

***NONE:** A PCFILE document cannot resolve to this type.

***EXN:** Specifies that PCFILE documents with the same extension are considered the same type as the type being defined.

***INFLR:** Specifies that PCFILE documents in a particular parent folder will be considered the same as the type being changed.

***VALUE:** Specifies that PCFILE documents that contain the specified string are considered the same as the type being defined.

***PGM:** Specifies that PCFILE documents will be considered as the type being defined based on the result of calling the specified

program. The program will return a flag indicating whether the document is this type.

EXN

Specifies the extensions that will identify a PCFILE of this type. There is an upper limit of 300 extensions. This parameter is valid only if the value of the PCFILEID keyword is *EXN.

extension: The document name extension value that identifies PCFILE documents as this type. An extension both uppercase and up to three characters in length.

Note: PCFILE documents sent from other AS/400 systems contain the name of the document from the originating system. This name is stored in the IDP of the distribution document and will be used for PCFILE identification.

INFLR

A list of folder names. If a document is in a folder with one of these names, it is considered of this type. This parameter is valid only if the value of the PCFILEID keyword is *INFLR.

parent-folder-name: The name of the parent folder where the PCFILE document is located. Folders are in uppercase.

VALUE

The character string that a document contains if it is this type. This parameter is valid only if the value of the PCFILEID parameter is *VALUE.

value: The hexadecimal string that uniquely identifies this document type. The string can be 1 to 16 bytes long.

starting-position: The starting position in the document where the specified string is found. The first byte in the document is identified as byte 0.

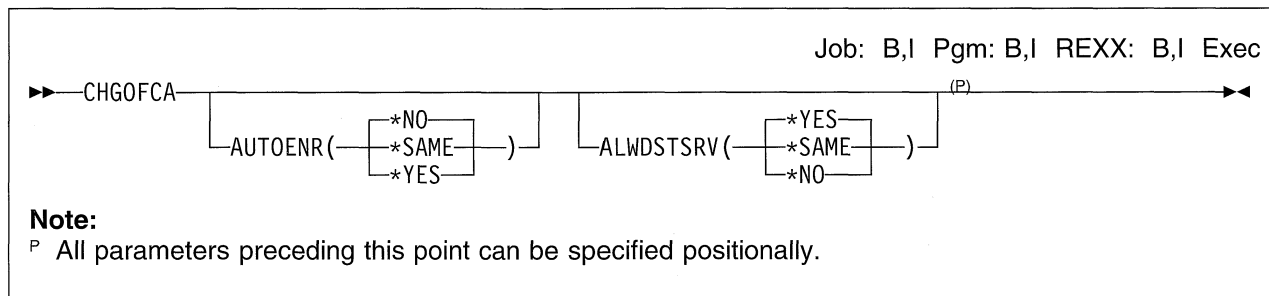
PGM

The name of a program that will determine if a PCFILE document is of this type. The parameter is valid only if the value of the PCFILEID is *PGM.

library-name: Specifies the name of the library where the user program is located.

program-name: Specifies the name of the user program.

CHGOFCA (Change Office Attributes) Command



Purpose

The Change Office Attributes (CHGOFCA) command changes the contents of the specified OfficeVision/400 value.

Restriction: You must have security administrator (*SECADM) authority to use this command.

Parameters

AUTOENR

Specifies whether users are automatically enrolled in the system distribution directory and in OfficeVision/400 when OfficeVision/400 services are requested.

***NO:** The user is not automatically enrolled when OfficeVision/400 services are requested. The user must be enrolled by a security administrator before OfficeVision/400 services can be used.

***SAME:** The value does not change.

***YES:** The user is automatically enrolled when OfficeVision/400 services are requested.

ALWDSTSRV

Specifies whether users are allowed to use distribution services commands when the MAIL option of OfficeVision/400 is not installed.

***YES:** The user is allowed to receive notes and messages.

***SAME:** The value does not change.

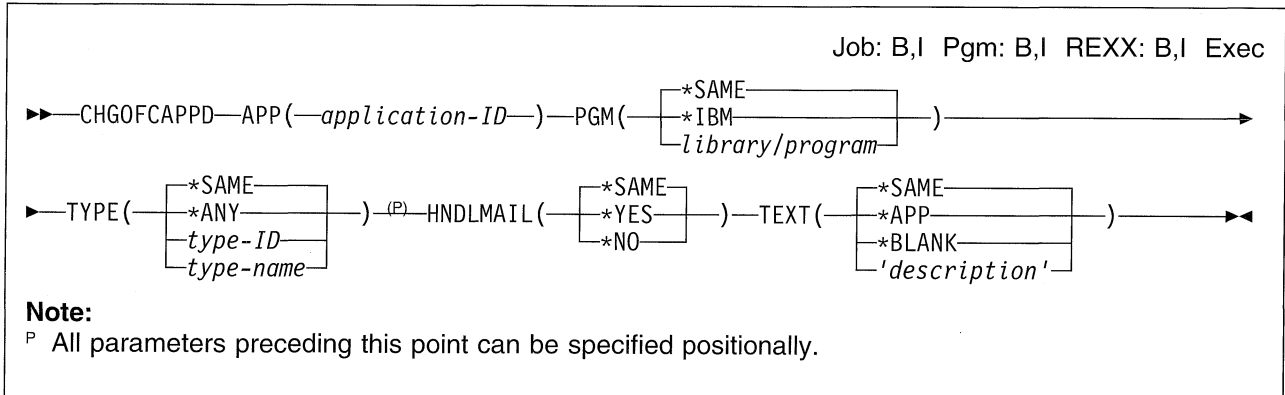
***NO:** The user is not allowed to receive notes and messages. If ALWDSTSRV(*NO) is specified when MAIL is installed, an error message is sent to the user and the value is set to *YES.

Example

```
CHGOFCA AUTOENR(*YES)
```

This command changes the OfficeVision/400 value to automatically enroll users when OfficeVision/400 services are requested.

CHGOFCAPPD (Change Office Application Description) Command



Purpose

The Change Office Application Description (CHGOFCAPPD) command allows the user to change the information associated with an office application.

Parameters

APP

The identifier of the application to change.

application-ID: The name of the application to change. This name must meet system object naming conventions.

PGM

The program called to run this application.

***SAME**: The current value is not changed.

***IBM**: The user OfficeVision/400 editor support is associated to this application.

library/program: The library containing the program that runs the application. There is no option to use any form of library list or current library resolution. The program must be explicitly library qualified.

TYPE

The input document type required by this application, if any.

***SAME**: The current value is not changed.

***ANY**: There is no type control for this application. Any type will be accepted.

type-ID: The type ID needed by the application. Any document this application is called

against will be converted to this type, or the reference will be handled as an error.

type-name: The type name of the type wanted by this application. Any document that this application is called against will be converted to this type, or the reference will be handled as an error.

HNDLMAIL

The mail functions are either supported or not supported by the application. If the mail document processing application does not support mail editing functions, performance improvements are possible.

***SAME**: The current value is not changed.

***YES**: This application supports the mail functions.

***NO**: This application does not support the mail functions.

TEXT

The descriptive text associated with the application definition.

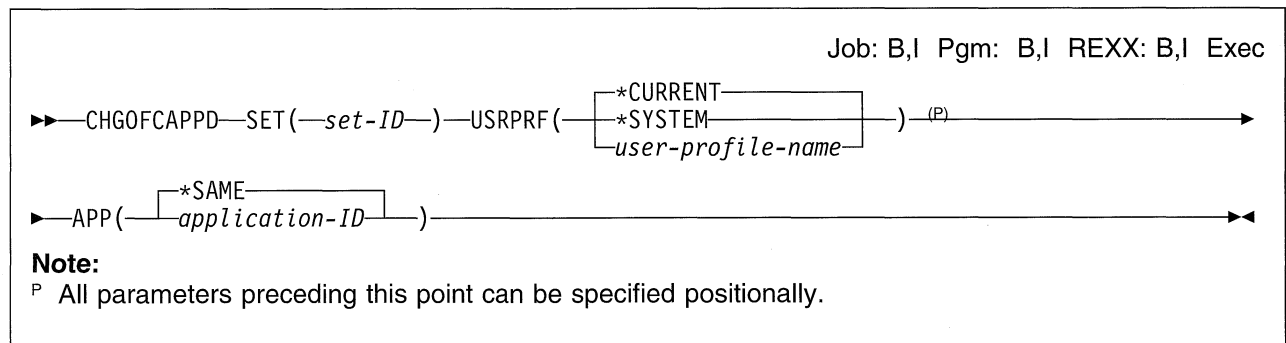
***SAME**: The current value is not changed.

***APP**: The text will be the same as the application ID.

***BLANK**: No text associated with the application.

'description': Up to 50 characters of text to be associated with the application. It will be kept in the input CCSID and converted if presented by a job supporting a different CCSID.

CHGOFCAPPE (Change Office Application Entry) Command



Purpose

The Change Office Application Entry (CHGOFCAPPE) command allows the user to change applications and sets for a specified user.

Parameters

SET

Specifies the set to be associated with the user.

set-ID: The ID of an existing set associated with an application.

USRPRF

Specifies the user for whom the office application entry is changed.

***CURRENT**: The application entries for the current signed on user are changed.

Note: If the user is not the security officer and does not have *SECADM authority, *CURRENT must be specified on this parameter.

***SYSTEM**: The system default values are used if the user has no requested application for a set.

user-profile-name: Specify the name of the user profile used.

APP

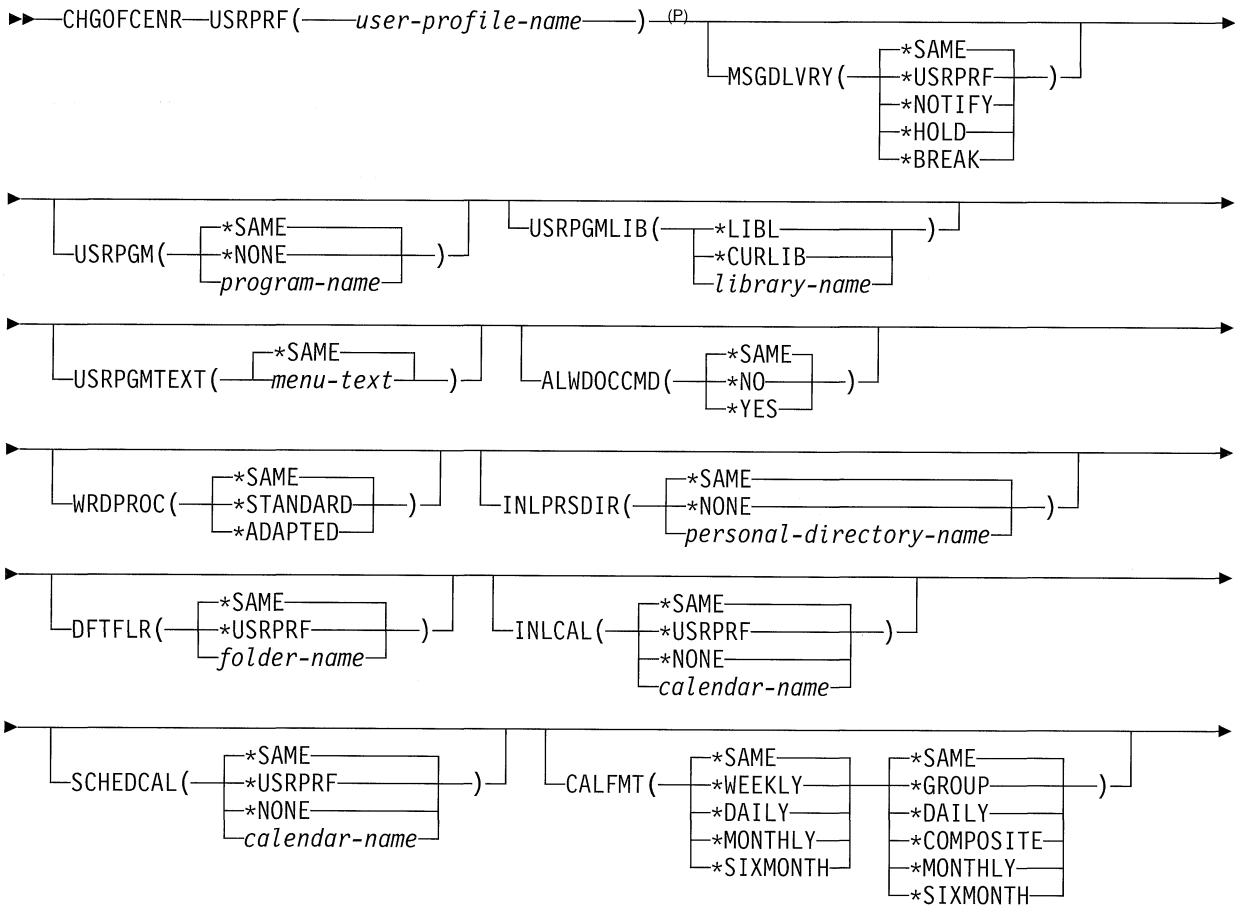
Specifies the identifier of the application being changed. The application name must conform to system object naming conventions.

***SAME**: The value does not change.

application-ID: Specify the ID of the application used.

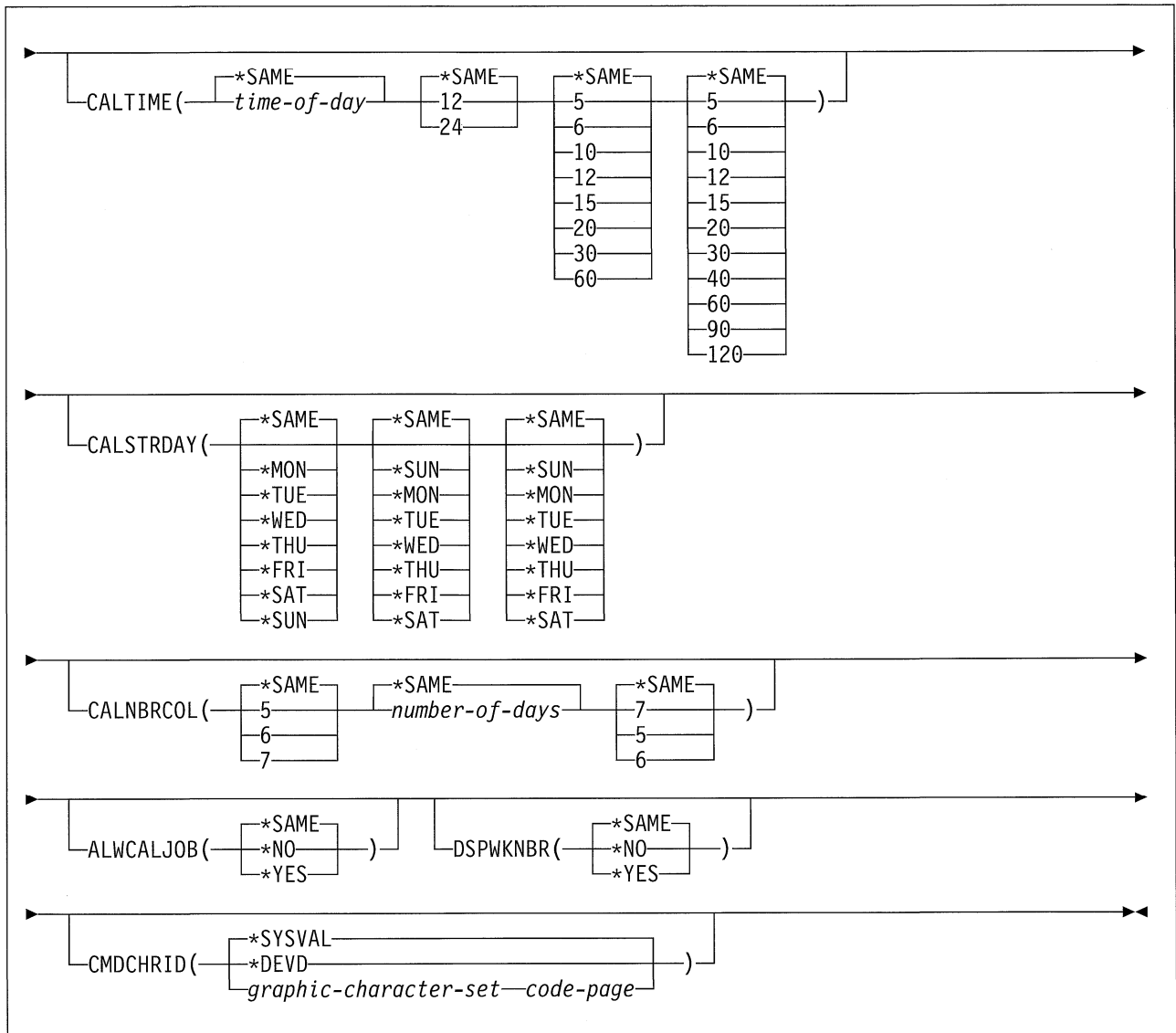
CHGOFCENR (Change Office Enrollment) Command

Job: B,I Pgm: B,I



Note:

^P All parameters preceding this point can be specified positionally.



Purpose

The Change Office Enrollment (CHGOFCENR) command allows the user to change a user's OfficeVision/400 enrollment information. Ignore the CAL and MAIL parameters when these options are not installed on the system.

Restriction: The user of this command must be designated a security officer and be enrolled in OfficeVision/400 as an administrator.

Parameters

USRPRF

Specifies the name of the user profile whose enrollment information is being changed.

user-profile-name: Specify the name of the user profile.

MSGDLVRY

Specifies how a user is notified of messages while using OfficeVision/400.

***SAME:** The delivery mode does not change.

***USRPRF:** The delivery mode is the same as the mode specified in the user profile.

***NOTIFY:** The audible alarm is sounded or the message waiting light is turned on. The light or alarm is turned off when the user looks at the message.

***HOLD:** Messages are held in the message queue until they are requested by the user or program.

***BREAK:** The user's job is interrupted and the message is displayed.

USRPGM

Returns the qualified name of the user program to be called when the user selects Option 50 on the OfficeVision/400 menu. No check is made to verify that the specified user program exists. The user program cannot require any parameters.

***SAME:** The user program does not change.

***NONE:** No user program is used.

program-name: Contains the name of the program that is called when the user selects option 50 from the OfficeVision/400 menu. The name must be from 1 to 10 characters long.

USRPGMLIB

Returns the qualified name of the library to be called when the user selects Option 50 on the OfficeVision/400 menu. No check is made to verify that the specified library exists.

***LIBL:** The library list is used to locate the user program.

***CURLIB:** The current library for the job is used to locate the current library for the job, QGPL is used.

library-name: Contains the name of the library the user program resides. The name can be from 1 to 10 characters long.

USRPGMTEXT

Returns the name of the text when the user selects Option 50 on the OfficeVision/400 menu.

***SAME:** The user text does not change.

menu-text: Contains the text that is shown when the user selects Option 50 on the OfficeVision/400 menu. The text can be a maximum of 40 characters.

ALWDOCCMD

Specifies whether the user can use control language (CL) commands when working with text documents.

***SAME:** The value specified for this parameter does not change.

***NO:** The user cannot use CL commands when working with text documents.

***YES:** The user can use CL commands when working with text documents.

WRDPROC

Specifies the type of word processing function used.

***SAME:** The type of word processing function used does not change.

***STANDARD:** The word processing function that is appropriate for the user's work station is used. If the user works on a personal computer, the PC text-assist function is used.

***ADAPTED:** The adapted word processing function is used, regardless of the type of work station.

INLPRSDIR

Specifies the name of the first personal directory the user will use when running personal directory functions. If ***NONE** is specified, the user must select a personal directory each time the personal directory function is used. No check is made to verify that the specified personal directory exists.

***SAME:** The name of the personal directory does not change.

***NONE:** No personal directory is used.

personal-directory-name: Specify the name of the personal directory.

DFTFLR

Specifies the name of the folder that is used whenever a folder is needed but not specified. If the folder does not exist, it is created.

***SAME:** The name of the default folder does not change.

***USRPRF:** The name of the folder is the same as the name of the user profile.

folder-name: Specify the name of the folder.

INLCAL

Specifies the name of the first calendar the user will see when using calendar functions. The name of a calendar, calendar group, or a distribution list calendar can be specified. If the specified calendar does not exist, it is created.

***SAME:** The name of the calendar does not change.

***USRPRF:** The name of the calendar is the same as the name of the user profile.

***NONE:** No calendar name is specified.

calendar-name: Specify the name of the calendar. The name can be a maximum of 10 characters.

SCHEDCAL

Specifies the name of the user's scheduling calendar. If the specified calendar does not exist, it is created.

***SAME:** The name of the calendar does not change.

***USRPRF:** The name of the calendar is the same as the name of the user profile.

***NONE:** No calendar name is specified.

calendar-name: Specify the name of the calendar. The name can be a maximum of 10 characters.

CALFMT

Specifies the type of view shown when only one calendar is used, and the type of view shown when more than one calendar is used.

Element 1: One-calendar View

***SAME:** The type of view shown does not change.

***WEEKLY:** A weekly calendar is shown.

***DAILY:** A daily calendar is shown. A daily calendar shows a detailed view of the calendar for a single day or several days.

***SIXMONTH:** A six-month calendar is shown. A six-month calendar shows six months on one display. It can be used to check dates, count the number of days from one date to another, or for any other task that requires a monthly format.

***MONTHLY:** A monthly calendar is shown. This calendar can be used to view the calendar items for one month.

Element 2: Multiple-calendar View

***SAME:** The type of view shown does not change.

***GROUP:** A group calendar is shown. A group calendar shows a single day with each of the calendars' schedules for that day displayed in a separate column.

***DAILY:** A daily calendar is shown. A daily calendar shows a detailed view of a single day with each of the calendars' schedules for that day displayed consecutively.

***COMPOSITE:** The available times on up to 10 calendars are shown in a five-day, six-day, or seven-day column format.

***SIXMONTH:** A six-month calendar is shown. A six-month calendar shows six months on one display. It can be used to check dates, count the number of days from one date to another, or for any other task that requires a monthly format.

***MONTHLY:** Specifies a monthly view of the calendar items.

CALTIME

Specifies the time of day at which the calendar starts, the format in which the time is shown to the user, and the number of minutes in each time slot on the calendar.

Element 1: Calendar Start Time

***SAME:** The time of day does not change.

time-of-day: Specify the time of day at which the calendar starts.

Element 2: Calendar Time Format

***SAME:** The time format does not change.

12: A 12-hour day is used.

24: A 24-hour day is used.

Element 3: Calendar Time Interval

***SAME:** The number of minutes in each time slot does not change.

30: Each time slot is 30 minutes long.

number-of-minutes: Specify the number of minutes in each time slot. Valid values are: 5, 6, 10, 12, 15, 20, 30, and 60.

Element 4: Monthly View Time Interval

***SAME:** The number of minutes in each time slot does not change.

60: Each time slot is 60 minutes long.

number-of-minutes: Specify the number of minutes in each time slot for the monthly calendar view. The valid values are 5, 6, 10, 12, 15, 20, 30, 40, 60, 90, and 120.

CALSTRDAY

Specifies the day of the week shown in the left column of the weekly calendar, and the day of the week shown in the left column of the six-month calendar.

Element 1: Weekly-view Calendar Start Day

***SAME:** The day of the week shown does not change.

***MON:** Monday is shown in the left column of the calendar.

***TUE:** Tuesday is shown in the left column of the calendar.

***WED:** Wednesday is shown in the left column of the calendar.

***THU:** Thursday is shown in the left column of the calendar.

***FRI:** Friday is shown in the left column of the calendar.

***SAT:** Saturday is shown in the left column of the calendar.

***SUN:** Sunday is shown in the left column of the calendar.

Element 2: Six-month Calendar Start Day

***SAME:** The day shown does not change.

***SUN:** Sunday is shown in the left column of the calendar.

***MON:** Monday is shown in the left column of the calendar.

***TUE:** Tuesday is shown in the left column of the calendar.

***WED:** Wednesday is shown in the left column of the calendar.

***THU:** Thursday is shown in the left column of the calendar.

***FRI:** Friday is shown in the left column of the calendar.

***SAT:** Saturday is shown in the left column of the calendar.

Element 3: Monthly View Start Day

***SAME:** The day shown does not change.

***SUN:** Sunday is shown in the left column of the calendar.

***MON:** Monday is shown in the left column of the calendar.

***TUE:** Tuesday is shown in the left column of the calendar.

***WED:** Wednesday is shown in the left column of the calendar.

***THU:** Thursday is shown in the left column of the calendar.

***FRI:** Friday is shown in the left column of the calendar.

***SAT:** Saturday is shown in the left column of the calendar.

CALNBRCOL

Specifies the number of columns that are shown on weekly, group, or composite views of the calendar. It also can specify the number of days that are shown on the daily calendar and the number of columns shown on the monthly view. Depending on the value specified on the CALFMT parameter, each column represents one of the following:

- A consecutive day for a weekly calendar
- A different user for the same day for a group calendar
- A consecutive day containing calendar items for up to 10 users for a composite calendar.

Element 1: Number of Columns

***SAME:** The number of columns does not change.

5: The number of columns is 5.

6: The number of columns is 6.

7: The number of columns is 7.

Element 2: Number of Days

***SAME:** The number of days shown does not change.

number-of-days: Specify the number of days shown on the daily calendar. Valid values range from 1 through 999.

Element 3: Number of Monthly View Columns

***SAME:** The number of columns does not change.

7: Seven columns are shown.

5: Five columns are shown.

6: Six columns are shown.

ALWCALJOB

Specifies whether the user can schedule jobs and procedures on calendars.

***SAME:** The value does not change.

***NO:** The user cannot schedule jobs and procedures.

***YES:** The user can schedule jobs and procedures.

DSPWK NBR

Specifies whether to display the week number.

***SAME:** The value does not change.

***NO:** Do not display the week number.

***YES:** Display the week number.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for data being entered as command parameter values. This character identifier is related to the display device used to specify the command. More information about CHRID processing is in the *Guide to Programming Displays*.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device are used.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

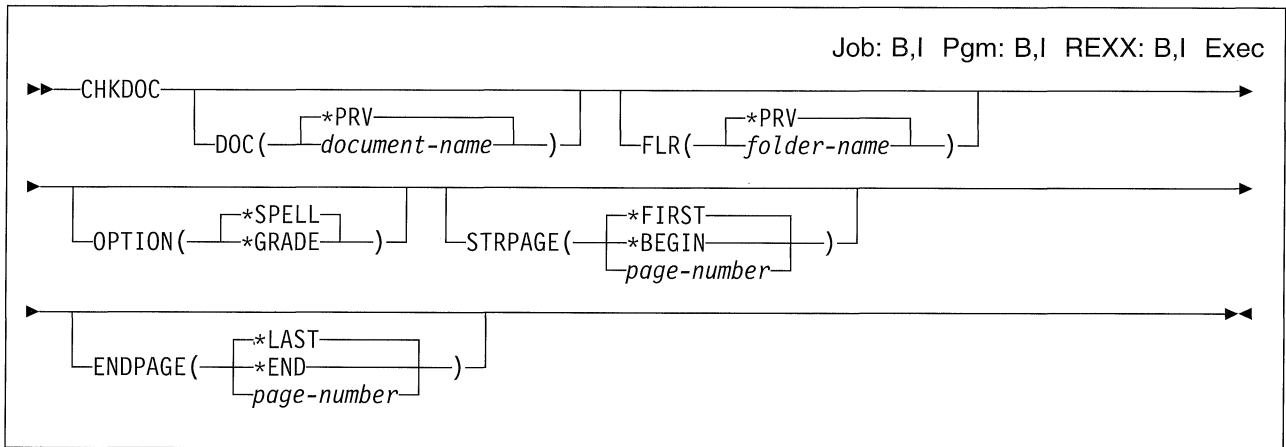
Example

```
CHGOFCENR USRPRF(JACKSON) WRDPROC(*STANDARD)
DFTFLR (JACKSON) INLCAL(JACKSON)
SCHEDCAL(JACKSON) CALTIME('08:00' 12 30)
CALSTRDAY(*MON) CALFMT(*WEEKLY)
```

This command changes the enrollment status of user profile JACKSON in OfficeVision/400 to:

- Standard word processing functions
- A default folder
- Calendar
- Personal calendar named JACKSON
- A Monday through Friday calendar displayed one day at a time that starts each day at 8 a.m. and has a slot for each 30-minute period in the 12-hour day

CHKDOC (Check Document) Command



Purpose

The Check Document (CHKDOC) command verifies the spelling of a document or performs a grade-level analysis on a document when using the word processing function of the OfficeVision/400 product. More information about checking documents is in the *Using OfficeVision/400* Word Processing*.

Parameters

DOC

Specifies the name of the document that is checked.

***PRV:** The name of the document used in the last session is checked for spelling or grade level.

document-name: Specify the name of the document to be checked for spelling or grade level.

FLR

Specifies the name of the folder that contains the document.

***PRV:** The name of the folder used in the last session is used.

folder-name: Specify the name of the folder that contains the document to be checked for spelling or grade level.

OPTION

Specifies the type of check that is performed.

***SPELL:** The spelling of the document is checked.

***GRADE:** The grade level of the document is checked.

STRPAGE

Specifies the starting page number for the check.

***FIRST:** The check starts on the first page of the document.

***BEGIN:** The check starts on the first page of the document.

page-number: Specify the number of the page of the document on which to start the check. Valid values range from 1.00 through 9999.99.

ENDPAGE

Specifies the ending page number for the check.

***LAST:** The check ends on the last page of the document.

***END:** The check ends on the last page of the document.

page-number: Specify the number of the page of the document on which the check ends. Valid values range from 1.00 through 9999.99.

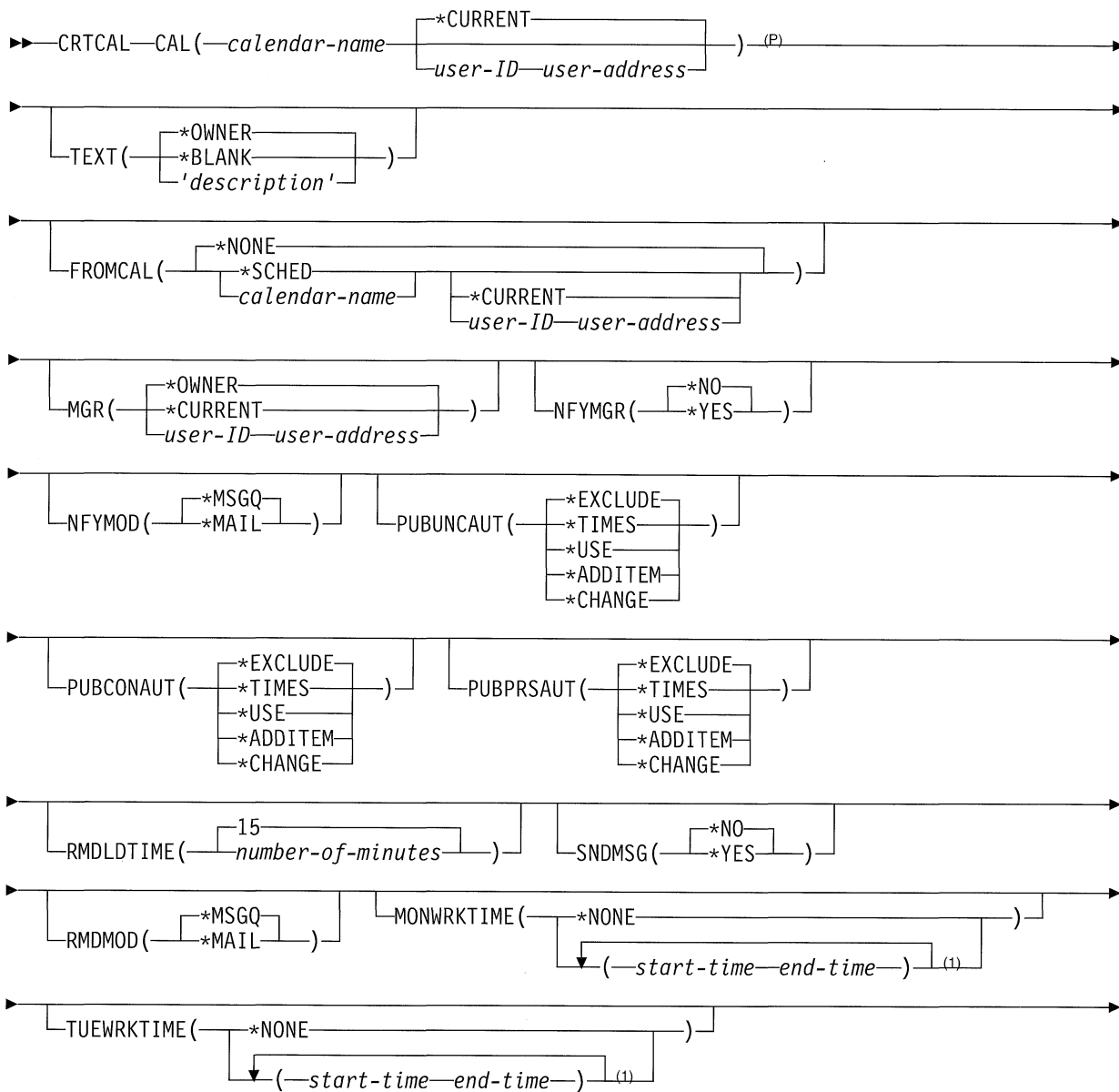
Example


```
CHKDOC DOC(MYDOC) FLR(MYFLR) OPTION(SPELL)  
STRPAGE(*BEGIN) ENDPAGE(*END)
```

This command checks the spelling of the document MYDOC in folder MYFLR from the first page to the last page.

CRTCAL (Create Calendar) Command

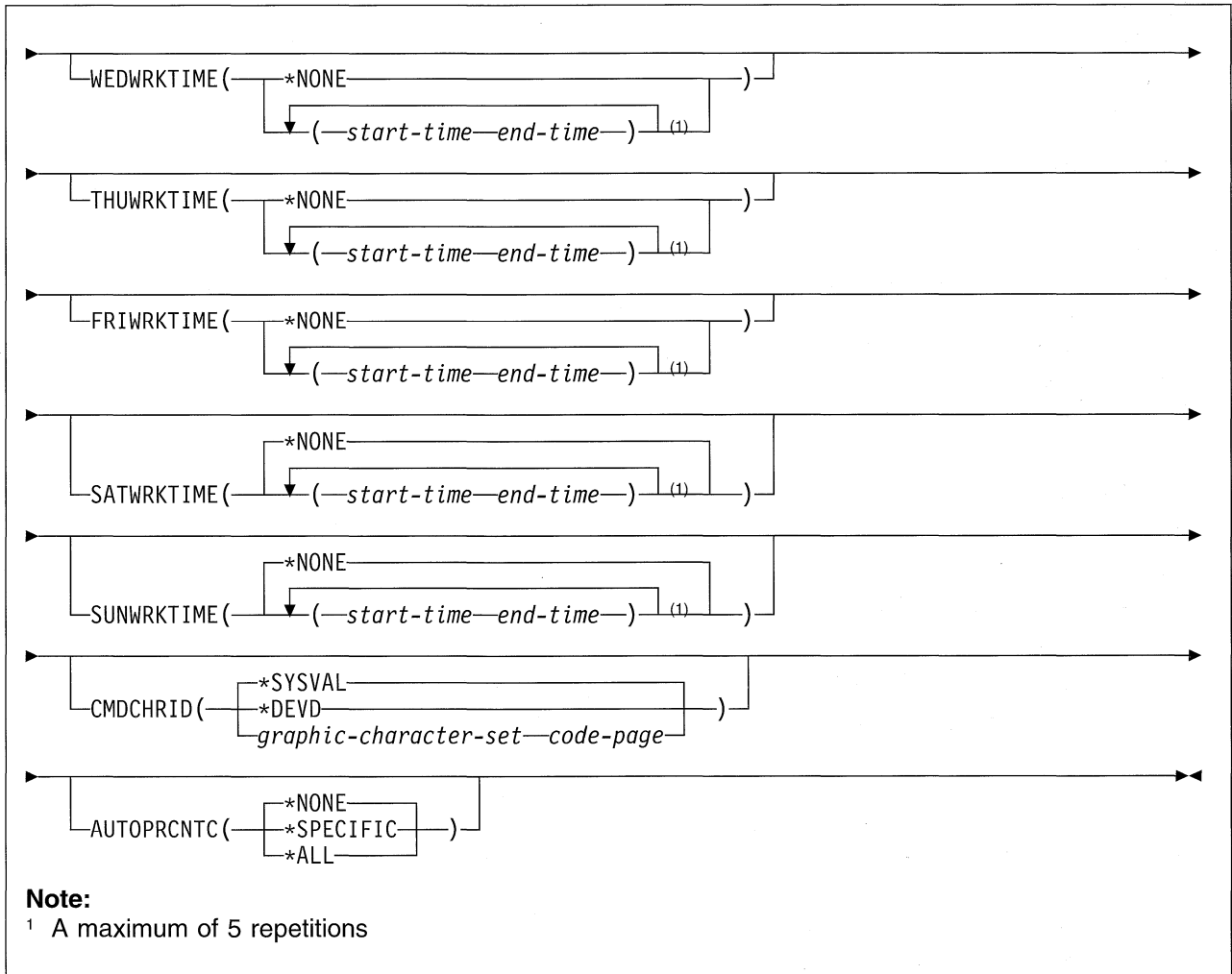
Job: B,I Pgm: B,I Exec



Notes:

¹ A maximum of 5 repetitions

^P All parameters preceding this point can be specified positionally.



Purpose

The Create Calendar (CRTCAL) command is used to create a new calendar. The owner, description, and manager of the calendar, the working times for the calendar, and the notification mode for items on the calendar can be specified.

Restrictions: The user must have *SECADM authority or *ALLOBJ authority to specify a user other than *CURRENT for the calendar owner.

Parameters

CAL

Specifies the name and owner of the calendar being created.

Element 1: Calendar Name

calendar-name: Specify the name of the calendar being created.

Element 2: Calendar Owner

***CURRENT:** The user issuing the command is the owner of the calendar being created.

user-ID: Specify the user ID of the owner of the calendar being created. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar being created. This value is valid only if user ID is also specified.

Note: The user of this command must have SECADM authority to specify a user other than *CURRENT.

TEXT

Specifies text that briefly describes the object. More information on this parameter is in "Appendix A. Expanded Parameter Descriptions".

CRTCAL

***OWNER:** The description of the calendar is the same as the description of the owner's system distribution directory entry.

***BLANK:** No text is used.

'description': Specify up to 50 characters of text.

FROMCAL

Specifies the name and owner of the calendar from which the values for the remaining parameters on this command are copied. If a calendar is specified on this parameter, values specified for any of the following parameters override the values from the copied calendar.

***NONE:** No values are copied from another calendar.

Element 1: Calendar Name

***SCHD:** The calendar to be copied from is the scheduling calendar of the owner specified on the second element of this parameter.

calendar-name: Specify the name of the calendar.

Element 2: Calendar Owner

***CURRENT:** The user issuing the command is the owner of the calendar.

user-ID: Specify the user ID of the owner of the calendar. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar. This value is valid only if user ID is also specified.

MGR

Specifies the name of the manager of the calendar being created.

***OWNER:** The manager of the calendar is the same as the owner of the calendar.

***CURRENT:** The user issuing the command is the manager of the calendar.

user-ID: Specify the user ID of the manager of the calendar. This value is valid only if user address is also specified.

user-address: Specify the user address of the manager of the calendar. This value is valid only if user ID is also specified.

NFYMGR

Specifies whether the manager of the calendar is notified of changes made to the calendar.

***NO:** The manager is not notified of changes to the calendar.

***YES:** The manager is notified of changes to the calendar.

NFYMOD

Specifies the notification mode for the manager of the calendar.

Note: This parameter is valid only if *YES is specified on the NFYMGR parameter.

***MSGQ:** The manager of the calendar receives a message when the calendar is changed.

***MAIL:** The manager of the calendar receives a mail item and a message when the calendar is changed.

PUBUNCAUT

Specifies the authority to unclassified calendar items that is given to users who do not have specific authority to the calendar or who are not on the authorization list.

***EXCLUDE:** The user cannot see unclassified calendar items.

***TIMES:** The user can see the times for which unclassified items on the calendar are scheduled, but cannot see any text associated with the items.

***USE:** The user can see the times for which unclassified items on the calendar are scheduled and the text associated with the items.

***ADDITEM:** The user is given *USE authority to the calendar. The user also can add items to the calendar and change the items added.

***CHANGE:** The user can change or delete unclassified calendar items. The user cannot delete the calendar itself or change the list of users authorized to the calendar.

PUBCONAUT

Specifies the authority to confidential calendar items that is given to users who do not have specific authority to the calendar or who are not on the authorization list.

***EXCLUDE:** The user cannot see confidential calendar items.

***TIMES:** The user can see the times for which the confidential items on the calendar are scheduled, but cannot see any text associated with the items.

***USE:** The user can see the times for which confidential items on the calendar are scheduled and the text associated with the items.

***ADDITEM:** The user is given *USE authority to the calendar. The user also can add items to the calendar and change the items added.

***CHANGE:** The user can change or delete confidential calendar items. The user cannot delete the calendar itself or change the list of users authorized to the calendar.

PUBPRSAUT

Specifies the authority to personal calendar items that is given to users who do not have specific authority to the calendar or who are not on the authorization list.

***EXCLUDE:** The user cannot see personal calendar items.

***TIMES:** The user can see the times for which personal items on the calendar are scheduled, but cannot see any text associated with the items.

***USE:** The user can see the times for which personal items on the calendar are scheduled and the text associated with the items.

***ADDITEM:** The user is given *USE authority to the calendar. The user also can add items to the calendar and change the items added.

***CHANGE:** The user can change or delete personal calendar items. The user cannot delete the calendar itself or change the list of users authorized to the calendar.

RMDLDTIME

Specifies the number of minutes before a calendar item is scheduled to occur that the owner is reminded of the item.

15: The owner is reminded 15 minutes before the item is scheduled to occur.

number-of-minutes: Specify the number of minutes before the item is scheduled to occur that the owner is reminded of the item.

SNDMMSG

Specifies whether a reminder is sent to the owner of the calendar before a calendar item

is scheduled to occur. The value specified on this parameter can be overridden for any particular item added to the calendar.

***NO:** A reminder is not sent.

***YES:** A reminder is sent.

RMDMOD

Specifies the notification mode for reminders sent to the owner of the calendar.

***MSGQ:** The owner of the calendar receives a message.

***MAIL:** The owner of the calendar receives a mail item and a message.

MONWRKTIME

Specifies the working time for Mondays. When this parameter is not specified, the working time is 0800 through 1700.

Element 1: Start Time

start-time: Specify the start time in 24-hour clock format. Up to 5 start times can be specified.

Element 2: End Time

end-time: Specify the end time in 24-hour clock format. Up to 5 end times can be specified.

Other Single Values

***NONE:** The owner of the calendar does not work on Mondays.

TUEWRKTIME

Specifies the working time for Tuesdays. When this parameter is not specified, the working time is 0800 through 1700.

Element 1: Start Time

start-time: Specify the start time in 24-hour clock format. Up to 5 start times can be specified.

Element 2: End Time

end-time: Specify the end time in 24-hour clock format. Up to 5 end times can be specified.

Other Single Values

***NONE:** The owner of the calendar does not work on Tuesdays.

WEDWRKTIME

Specifies the working time for Wednesdays. When this parameter is not specified, the working time is 0800 through 1700.

Element 1: Start Time

start-time: Specify the start time in 24-hour clock format. Up to 5 start times can be specified.

Element 2: End Time

end-time: Specify the end time in 24-hour clock format. Up to 5 end times can be specified.

Other Single Values

***NONE:** The owner of the calendar does not work on Wednesdays.

THUWRKTIME

Specifies the working time for Thursdays. When this parameter is not specified, the working time is 0800 through 1700.

Element 1: Start Time

start-time: Specify the start time in 24-hour clock format. Up to 5 start times can be specified.

Element 2: End Time

end-time: Specify the end time in 24-hour clock format. Up to 5 end times can be specified.

Other Single Values

***NONE:** The owner of the calendar does not work on Thursdays.

FRIWRKTIME

Specifies the working time for Fridays. When this parameter is not specified, the working time is 0800 through 1700.

Element 1: Start Time

start-time: Specify the start time in 24-hour clock format. Up to 5 start times can be specified.

Element 2: End Time

end-time: Specify the end time in 24-hour clock format. Up to 5 end times can be specified.

Other Single Values

***NONE:** The owner of the calendar does not work on Fridays.

SATWRKTIME

Specifies the working time for Saturdays.

***NONE:** The owner of the calendar does not work on Saturdays.

Element 1: Start Time

start-time: Specify the start time in 24-hour clock format. Up to 5 start times can be specified.

Element 2: End Time

end-time: Specify the end time in 24-hour clock format. Up to 5 end times can be specified.

SUNWRKTIME

Specifies the working time for Sundays.

***NONE:** The owner of the calendar does not work on Sundays.

Element 1: Start Time

start-time: Specify the start time in 24-hour clock format. Up to 5 start times can be specified.

Element 2: End Time

end-time: Specify the end time in 24-hour clock format. Up to 5 end times can be specified.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device is used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify the graphic character set.

Element 2: Code Page

code-page: Specify the code page.

AUTOPRCNTC

Specifies whether meeting notices are automatically processed for the user's calendar.

***NONE:** All meeting notices must be processed manually.

***SPECIFIC:** Meeting notices are automatically processed when the user scheduling the meeting is authorized to the calendar.

***ALL:** All meeting notices are automatically processed.

Examples

Example 1: Creating a Calendar with all Defaults

```
CRTCAL CAL(RMSMITH)
```

This command creates a calendar with all the defaults. The user issuing the command is the owner and manager of the calendar. The public has no authority to the items on the calendar. No notifications are sent to the manager or the owner. The working times are 8 to 5, Monday through Friday.

Example 2: Creating a Calendar with a Different Manager

```
CRTCAL CAL(RMSMITH) TEXT('My calendar')
MGR(SEC44B RCHAS1) PUBUNCAUT(*USE)
WEDWRKTIME(*NONE)
```

This command creates a calendar with the user issuing the command as the owner. The manager of the calendar is SEC44B. The public has authority to look at unclassified items on the calendar. The public has the authority to see only the times of confidential and personal items. Working times are 8 to 5 on Monday, Tuesday, Thursday, and Friday. There are no working times on Wednesday, Saturday, and Sunday.

Example 3: Creating a Calendar for a Conference Room

```
CRTCAL CAL(CONFROOM SEC44B RCHAS1)
TEXT('Calendar for the room')
PUBUNCAUT(*ADDITEM) RMDLDTIME(30)
SNDMSG(*YES) SATWRKTIME(('8:00' '17:00'))
SUNWRKTIME(('8:00' '17:00'))
```

This command creates a calendar for a conference room. The calendar is owned by SEC44B. The public is authorized to add unclassified items to the calendar. unclassified items to the calendar. The public has the authority to see only the times of confidential and personal items. The owner, SEC44B, is sent a reminder message 30 minutes before each item on the calendar. The conference room is available from 8 to 5, 7 days a week.

CRTDOCSET (Create Document Set) Command

Job: B,I Pgm: B,I REXX: B,I Exec

▶ CRTDOCSET SET (—*set-ID*—) ^(P) TEXT (— *SET
*BLANK
'description' —) ▶

Note:
^P All parameters preceding this point can be specified positionally.

Purpose

The Create Document Set (CRTDOCSET) command allows the user to create new document sets on the system.

Parameters

SET

The name of the set being created. The name must conform to system object naming conventions.

set-ID: The name of the set created.

TEXT

The text description to be assigned to this set.

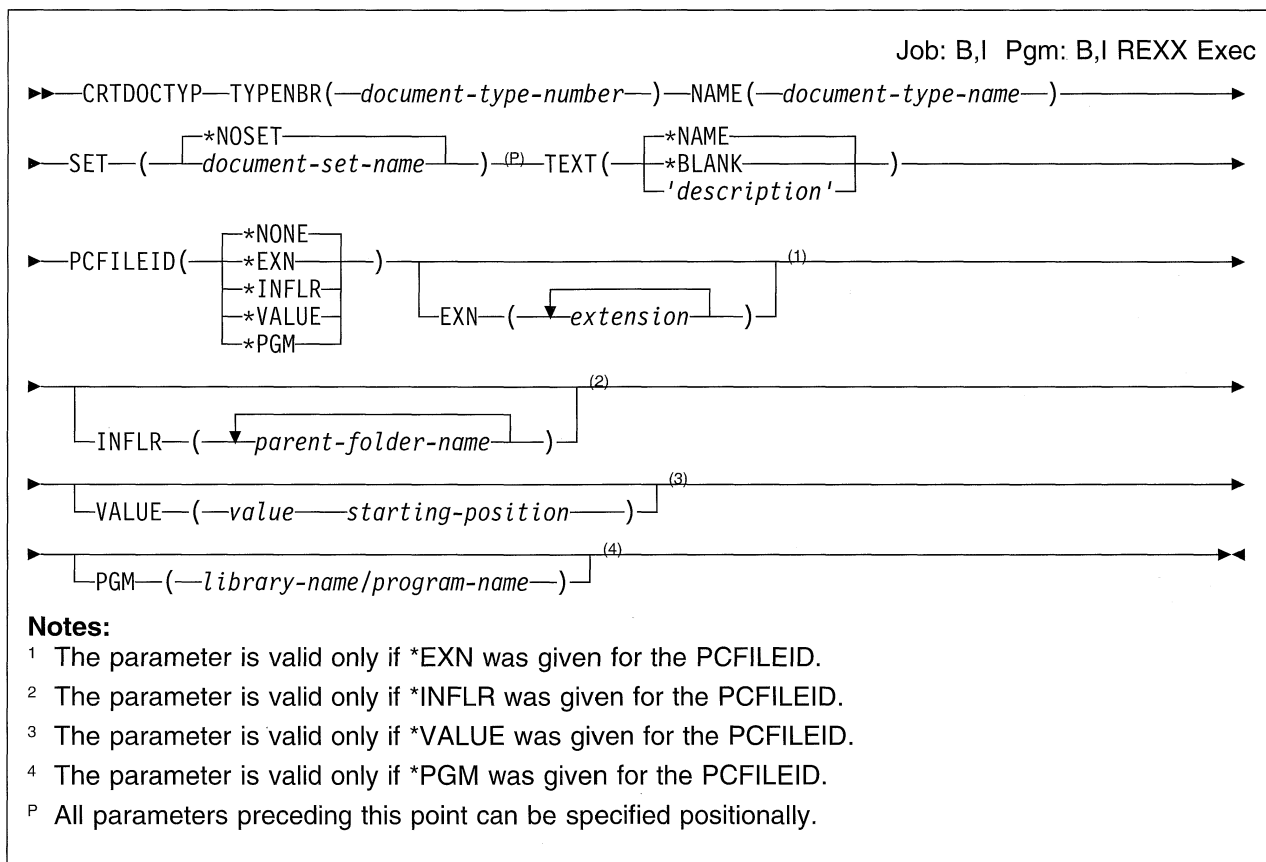
***SET:** The text will be identical on the set name.

***BLANK:** No text will be specified.

'description': The description to be assigned.

It will be in the input CCSID and converted if presented by a job supporting a different CCSID.

CRTDOCTYP (Create Document Type) Command



Purpose

The Create Document Type (CRTDOCTYP) command allows the user to add a new document type definition to the system and to identify the set associated with the new type. As part of the type definition, the user can specify the values for changing PCFILES to this new type.

Parameters

TYPENBR

Specifies the number of the document type being defined.

document-type-number: The numeric value between 32768 and 65535 that is associated with the DIA document type being defined.

Note: Values from 1-32767 are reserved for system use. Values above 32767 are available for use.

NAME

Specifies the name to be associated with the type.

document-type-name: The alphanumeric name to be associated with this document type. The name is 1-10 characters, it must meet the system object naming conventions, and it must be unique.

Note: The names of IBM-supplied document types will be translated for each supported language. Because of this, the uniqueness of a new name is assured only with other user-assigned names within the currently used language. It is possible to assign a name that will conflict with that of another language. When another language is in use, the name will refer to the IBM-supplied type.

SET

Specifies the set to which the type belongs.

***NOSET:** This type belongs to no set.

document-set-name: The name of an existing set to which this type belongs.

TEXT

Specifies a text description assigned to this type. The description can be 1 to 50 characters long.

***NAME:** The text will be identical to the name assigned to the type.

***BLANK:** No text will be specified.

'description': The assigned description. It will be kept in the input CCSID and converted if presented by a job supporting a different CCSID.

PCFILEID

Specifies the identification values for changing PCFILE type documents. These documents will be the document type specified on TYPENBR. The values used are the document name extension, parent folder name, a data string within the document, or the result of calling an external program.

***NONE:** A PCFILE document cannot resolve to this type.

***EXN:** Specifies that PCFILE documents with the name extension are considered the same as the type being defined.

***INFLR:** PCFILE documents that are directly contained in the folder with the name specified are considered the same as the type being defined.

***VALUE:** Specifies that PCFILE documents that contain the specified string are the same as the type being defined.

***PGM:** Specifies that PCFILE documents will be considered the same as the type being defined based on the result of calling the specified program. The program will return a flag indicating whether the document is this type.

EXN

The extensions that will identify a PCFILE of this type. There is an upper limit of 300 extensions. This parameter is valid only if the value of the PCFILEID keyword is *EXN.

extension: The document name extension value that identifies PCFILE documents as this type. An extension is up to three characters long and is uppercase.

Note: PCFILE documents sent from other AS/400 systems contain the name of the document from the originating system. This name is stored in the IDP of the distribution document and will be used for PCFILE identification.

INFLR

A list of folder names. If a document is in a folder with one of these names, it is of this type. This parameter is valid only if the value of the PCFILEID keyword is *INFLR.

parent-folder-name: The name of the parent folder in which the PCFILE document is located. Folder names are in uppercase.

VALUE

The character string that a document contains if it is this type. This parameter is valid only if the value of the PCFILEID parameter is *VALUE.

value: The hexadecimal string that uniquely identifies this document type. It can be 1 to 16 bytes long.

starting-position: The starting position in the document where the specified string is found with the first byte in the document is identified as byte 0.

PGM

The name of a program that will determine if a PCFILE document is of this type. The parameter is valid only if the value of the PCFILEID is *PGM.

library-name: Specifies the name of the library where the user program is located.

program-name: Specifies the name of the user program. See "PCFILE User Program Interface (PCFI0000)" on page 11-19 for the program parameters.

Example

```
CRTDOCTYP TYPNBR(32775) NAME(STDSS) SET(SPDSHEET)
TEXT('Standard spreadsheet') PCFILEID(*EXN)
EXN(WKS WKQ)
```

This example defines the DIA document type 32775 as a "Standard spreadsheet" with a name

- | of STDSS. It is a member of the document set
- | named (SPDSHEET). PCFILE documents will be
- | changed to this document type if they have an
- | extension of WKS or WKQ.

CVTDOC (Convert Document) Command

Job: B,I Pgm: B,I REXX Exec

```

| ▶▶ CVTDOC DOC (—document-name—) FLR (—folder-name—) TYPE (—type-ID—) —(P)▶▶
|                                     |type-name|
|
| ▶ CVTCONTENT (— *YES — *NO —) ▶▶
|
| Note:
| P All parameters preceding this point can be specified positionally.

```

Purpose

The Convert Document (CVTDOC) command allows the user to convert a document in order to change the type or use the registered conversion routines to convert the contents of a document.

The CVTDOC command supports single-step conversions. If multi-step or iterative conversions are needed, new conversions are done by performing several single-step CVTDOC conversions. One exception to the single-step conversion rule is for RFTAS400 and FFTAS400. The conversion for these types to RFTDCA and FFTDCA will be automatically called before calling the user conversion program.

Parameters

DOC

Specifies the name of the document to convert.

document-name: The name of the document.

FLR

Specifies the name of the folder containing the document to convert.

folder-name: The folder path leading from the root to the folder containing the document to convert.

TYPE

Specifies the type to which the document is converted.

type-ID: Specify the ID of the type to which the document is converted.

type-name: Specify the name of the type to which the document is converted.

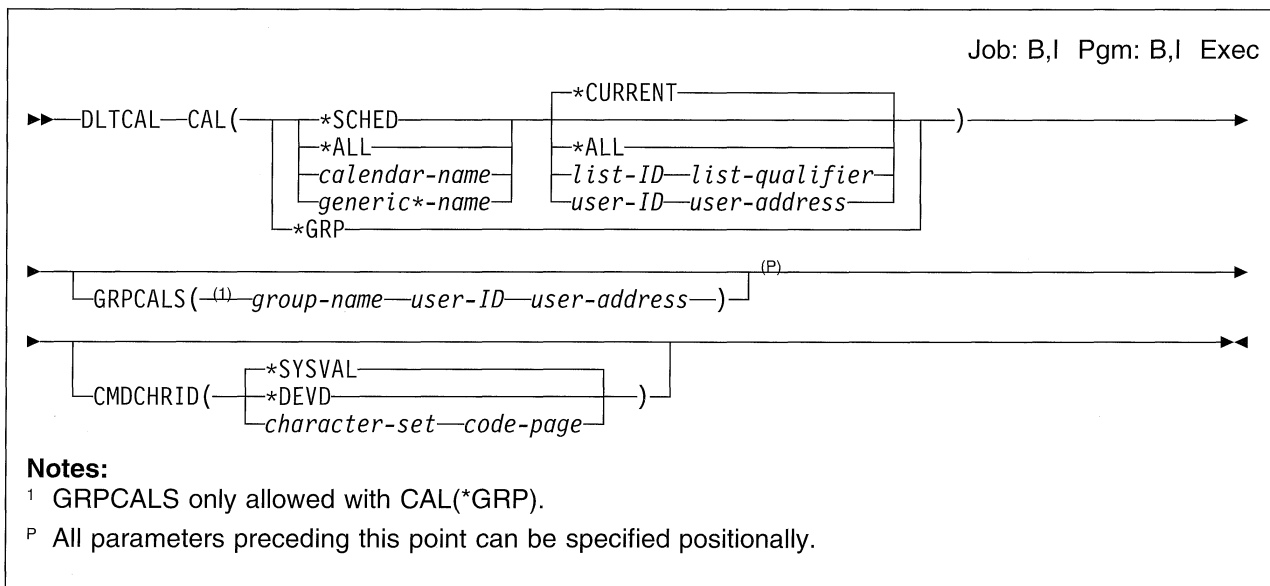
CVTCONTENT

Specifies whether the contents of the document are processed using the conversion routines or only the type is changed.

***YES:** The contents of the document are converted.

***NO:** The contents of the document are not converted. Only the type of the document is changed.

DLTCAL (Delete Calendar) Command



Purpose

The Delete Calendar (DLTCAL) command is used to delete one or more calendars from a local system. When a calendar is deleted, all of the items on the calendar are also deleted.

Restriction: The user must have *SECADM special authority, *ALLOBJ special authority, or be the owner of a calendar to delete it.

Parameters

CAL

Specifies the name and owner of the calendar being deleted. A calendar name and owner name, or *GRP, can be specified on this parameter.

Element 1: Calendar Name

***SCHED:** Scheduling calendars owned by the users specified on Element 2 of this parameter are deleted.

***ALL:** All calendars owned by the users specified on Element 2 of this parameter are deleted.

Note: The user must have *SECADM or *ALLOBJ authority to specify *ALL for both the calendar name and the

owner. If CAL(*ALL *ALL) is specified, no users can be using the calendar function when this command is run.

calendar-name: Specify the name of the calendar being deleted.

generic-name:* Specify the generic name of the calendar. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all calendars with names that begin with the generic name, and for which the user has authority, are deleted. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete calendar name.

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify 'MIKE**' to delete all calendars that have names beginning with 'MIKE*'.
Element 2: Calendar Owner

***CURRENT:** Calendars owned by the user issuing the command are deleted.

***ALL:** All named calendars owned by local users are deleted.

Note: The user must have *SECADM authority to specify *ALL.

list-identification: Specify a distribution list. All named calendars owned by users on the distribution list are deleted.

user-ID: Specify the user ID of the owner of the calendar being deleted. This value is valid only if user address is also specified.

address: Specify the user address of the owner of the calendar being deleted. This value is valid only if user ID is also specified.

Other Single Value:

***GRP:** All of the calendars in the calendar group specified on the GRPCALS parameter are deleted.

Note: If *GRP is specified, the GRPCALS parameter must also be specified.

GRPCALS

Specifies a local calendar group to be deleted. This parameter is valid only if *GRP is specified on the CAL parameter. A value for Element 1 must be specified if GRPCALS is specified. When the calendars in a group are deleted, the calendar group itself is not deleted.

Element 1: Group Name

group-name: Specify the local calendar group containing the calendars to be deleted.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group.

Element 3: User Address

user-address: Specify the user address of the owner of the calendar group.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command

parameter values. This parameter applies to the CAL parameter and the GRPCALS parameter.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device are used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples

Example 1: Deleting a Scheduling Calendar

```
DLTCAL CAL(*SCHED)
```

This command deletes the scheduling calendar of the user issuing the command.

Example 2: Deleting All Calendars

```
DLTCAL CAL(*ALL)
```

This command deletes all the calendars owned by the user issuing the command.

Example 3: Deleting All Calendars in a Group

```
DLTCAL CAL(*GRP) GRPCALS(DEPT54P)
```

This command deletes all calendars in the DEPT54 calendar group. The group is not deleted, but it is now empty.

DLTDOCSET (Delete Document Set) Command

Job: B,I Pgm: B,I REXX: B,I Exec

▶—DLTDOCSET—SET(—*set-ID*—)—^(P)DLTDOCTYP(—*NO
*YES—)—▶

Note:

^P All parameters preceding this point can be specified positionally.

Purpose

The Delete Document Set (DLTDOCSET) command allows the user to delete an existing document set.

Parameters

SET

Specifies the name of the set to be deleted.

set-ID: The name of the set to be deleted.

DLTDOCTYP

Indicates if the types contained in the set are to be deleted from the system. This delete will be done the same as DLTDOCTYP. The use of any type in this set as the input type for an application will cause the deletion to fail.

***NO:** Do not delete the types.

***YES:** Delete the types associated with the set.

DLTDOCTYP (Delete Document Type) Command

| | |
|--|--|
| | Job: B,I Pgm: B,I REXX: B,I Exec |
| <pre> DLTDOCTYP—TYPE (—<i>type-ID</i>—) —(P) —<i>type-name</i>— </pre> | <p>Note: ^P All parameters preceding this point can be specified positionally.</p> |

Purpose

The Delete Document Type (DLTDOCTYP) command allows the user to delete an existing type definition from the system. Deleting a type will automatically delete any set membership active for the type.

Note: A document type referenced in an office application description will not be deleted.

Parameters

TYPE

Specifies the name or number of the document type being deleted.

type-ID: The numeric value between 1 and 65535 that is associated with the DIA document type.

Note: Values from 1-32767 are reserved for system use. Valid numeric range from 32768 to 65535 are available for use.

type-name: The name of the type being deleted.

DLTOFCAPPD (Delete Office Application Description) Command

Job: B,I Pgm: B,I REXX Exec

▶—DLTOFCAPPD—APP(*—application-ID—*)(P)—————▶◀

Note:

^P All parameters preceding this point can be specified positionally.

Purpose

The Delete Office Application Description (DLTOFCAPPD) command allows the user to remove an office application from the system.

Parameters

APP

Specifies the name of the application to delete. The application name must conform to system object naming conventions.

application-ID: The name of the application to be deleted.

Example

DLTOFCAPPD APP(MSGALL)

This command removes the office application MSGALL from the system.

DLTOUTMAIL (Delete Outgoing Mail) Command

| |
|--|
| Job: B,I Pgm: B,I REXX: B,I Exec |
| ▶▶—DLTOUTMAIL—DAYS(<i>—number-of-days—</i>)—▶▶ |

Purpose

The Delete Outgoing Mail (DLTOUTMAIL) command allows the user to delete outgoing mail items for distributions in which confirmation of delivery was not requested. The user can use this command to clean up files that contain old outgoing mail.

Restrictions: The user must have *ALLOBJ special authority to use this command.

Parameters

DAYS

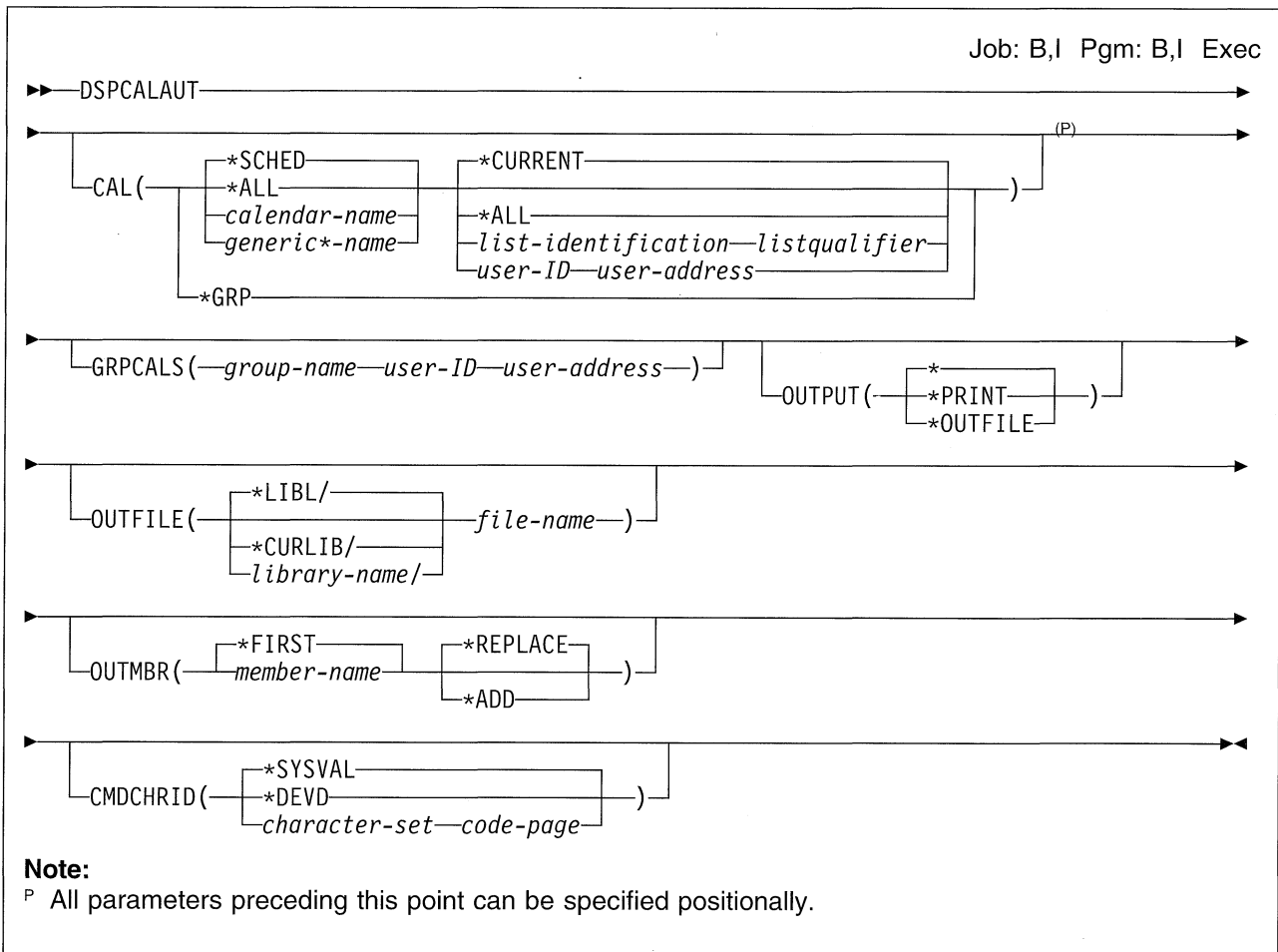
Specifies the number of days that outgoing mail can accumulate on the system before it is deleted.

Example

```
DLTOUTMAIL DAYS(14)
```

This command deletes outgoing mail after it has been on the system for 14 days.

DSPCALAUT (Display Calendar Authority) Command



Purpose

The Display Calendar Authority (DSPCALAUT) command is used to display the public and private authorities for a local calendar. All the private authorities are displayed if the user is the owner of the calendar, has *SECADM special authority, or *ALLOBJ special authority. If not, only the current user's private authority to the calendar and the public authority are displayed. The authorities can be displayed, printed, or directed to a database file.

Parameters

CAL

Specifies the name and owner of the calendar for which the authorities are to be displayed. A calendar name and owner name, or *GRP, can be specified on this parameter.

Element 1: Calendar Name

***SCHED:** The authorities for the scheduling calendars owned by the users specified in Element 2 of this parameter are displayed.

***ALL:** The authorities for all calendars owned by the users specified on Element 2 of this parameter are displayed.

calendar-name: Specify the name of the calendar for which the authorities are displayed.

generic-name:* Specify the generic name of the calendar for which the authorities are displayed.

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a

generic name. For example, specify 'MIKE**' to display authorities for all calendars that have names beginning with 'MIKE*'.

Element 2: Calendar Owner

***CURRENT:** The authorities for the calendar or calendars owned by the user of the command are displayed.

***ALL:** The authorities for the calendars that have the name specified on Element 1 of this parameter and are owned by local users are displayed.

list-identification: Specify a distribution list. The authorities for calendars specified on Element 1 of this parameter and are owned by users on the distribution list are displayed.

user-ID: Specify the user ID of the owner of the calendar or calendars to be displayed. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar or calendars to be displayed. This value is valid only if user ID is specified.

Other Single Values:

***GRP:** The authorities for the calendars in the calendar group specified on the GRPCALS parameter are displayed.

GRPCALS

Specifies the local calendar group that is displayed. This parameter is valid only if CAL(*GRP) is also specified. A value for Element 1 must be specified if GRPCALS is specified.

Element 1: Group Name

group-name: Specify the local calendar group for which authorities are displayed.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group for which authorities are displayed.

Element 3: User Address

user-address: Specify the user address of the owner of the calendar group for which authorities are displayed.

OUTPUT

Specifies whether the output from this command is displayed, printed, or directed to a database file.

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

***PRINT:** The output is printed with the job's spooled output.

***OUTFILE:** The output is directed to a database file.

OUTFILE

Specifies the qualified name of the database file to which the output of this command is directed. If the file does not exist, the system creates a file in the specified library. If a new file is created, system file QAOYCALAUT in system library QOFC with a record format name of CALAUT is used as a model. If the file already exists, it must have this format.

***LIBL:** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file is created in the QGPL library.

***CURLIB:** The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the file is located.

file-name: Specify the name of the output file.

OUTMBR

Specifies the name of the database file member that receives the output of this command.

Element 1: Member to Receive Output

***FIRST:** The first member in the file receives the output. If it does not exist, the system creates a member with the same name as the file specified on the OUTFILE parameter.

member-name: Specify the name of the file member that receives the output. If it does not exist, the system creates it.

Element 2: Operation to Perform on Member

DSPCALAUT

***REPLACE:** The output data replaces existing records in the specified member.

***ADD:** The output data is added after existing records in the specified member.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values. This parameter applies to the CAL parameter and the GRPCALS parameter.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device is used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Example

Example 1: Displaying Authorities for Scheduling Calendar

```
DSPCALAUT
```

This command displays the authority information for the scheduling calendar of the user issuing the command.

Example 2: Printing authorities for Scheduling Calendar

```
DSPCALAUT CAL(*SCHED PJC) OUTPUT(*PRINT)
```

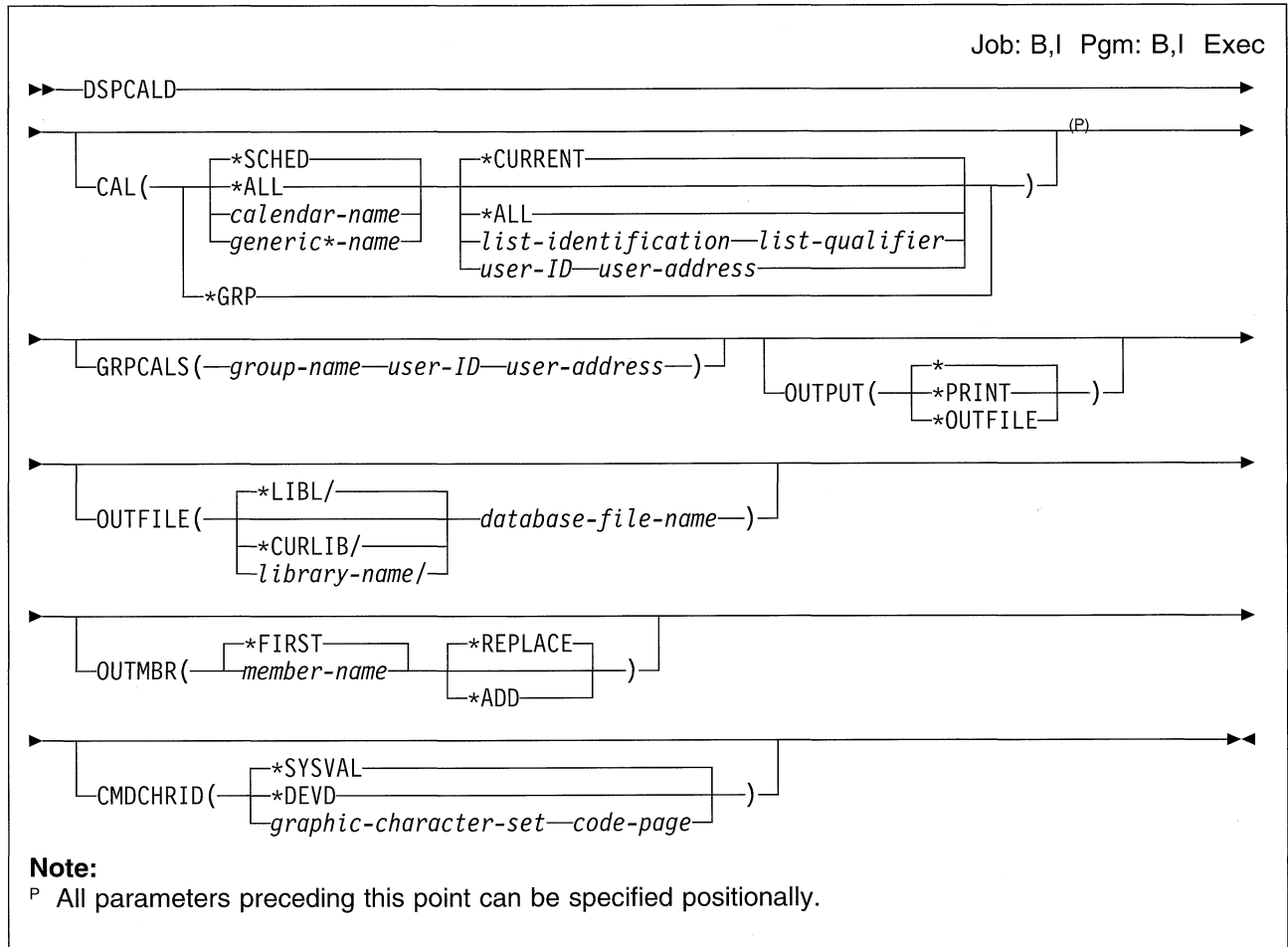
This command prints the authority information for the scheduling calendar of PJC.

Example 3: Generating An Authority Database File

```
DSPCALAUT CAL(CONFRM1 SEC44B RCHAS1)
  OUTPUT(*OUTFILE) OUTFILE(RMSLIB/CALAUT)
  OUTMBR(*FIRST *REPLACE)
```

This command generates a database file containing the authority information for calendar CONFRM1.

DSPCALD (Display Calendar Description) Command



Purpose

The Display Calendar Description (DSPCALD) command is used to display information about a local calendar. The information can be displayed, printed, or directed to a database file.

The information displayed includes the name, owner, and description of the calendar, the name of the manager of the calendar, the working times of the calendar, the notification mode for calendar items, and the authorities to calendar items granted to calendar users.

If the user is the owner of the calendar, has *SECADM special authority, or *ALLOBJ authority, the *PUBLIC authority and all of the private authorities to calendar items are displayed. If the user is not the owner and does not have

*SECADM or *ALLOBJ authority, only the private authority of the user issuing the command and the *PUBLIC authority are displayed. A user with *EXCLUDE authority sees only calendar name, description, owner and authorities. A user with *TIMES authority sees the calendar name, description, owner, working times and authorities.

If output is printed or sent to a file, only the public and current user's authority is included.

Parameters

CAL

Specifies the name and owner of the calendar for which the description is displayed. A calendar name and owner name, or *GRP, can be specified on this parameter.

Element 1: Calendar Name

***SCHED:** Calendar descriptions for the scheduling calendars owned by the users specified on the second element of this parameter are displayed.

***ALL:** Descriptions for all calendars owned by the users specified on the second element of this command are displayed.

calendar-name: Specify the name of the calendar for which the description is displayed.

generic-name:* Specify the generic name of the calendar for which a description is displayed. A generic name is a character string that contains one or more characters followed by an asterisk (*). Calendar descriptions for calendars that have names that match the generic name and are owned by the users specified on the second element of this parameter are displayed.

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify 'MIKE**' to display the descriptions of calendars that have names beginning with 'MIKE*'.

Element 2: Calendar Owner

***CURRENT:** Calendar description information for calendars owned by the user issuing the command is displayed.

***ALL:** Calendar description information for all calendars that have the name specified on the first element of this parameter and are owned by local users are displayed.

list-identification: Specify a distribution list. Calendar descriptions for calendars that have the name specified on the first element of this parameter and that are owned by users on the distribution list are displayed.

user-ID: Specify the user ID of the owner of the calendar for which the description is displayed. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar for which the description is displayed. This value is valid only if user ID is also specified.

Other Single Values

***GRP:** Calendar descriptions for the calendars in the calendar group specified on the GRPCALS parameter are displayed.

Note: If *GRP is specified, the GRPCALS parameter must also be specified.

GRPCALS

Specifies the local calendar group for which descriptions are displayed.

Note: This parameter is required if *GRP is specified on the CAL parameter.

A value for Element 1 must be specified if GRPCALS is specified.

Element 1: Group Name

group-name: Specify the local calendar group for which descriptions are displayed.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group for which descriptions are displayed.

Element 3: User Address

user-address: Specify user address of the owner of the calendar group for which descriptions are displayed.

OUTPUT

Specifies whether the output from this command is displayed, printed, or directed to an database file.

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

***PRINT:** The output is printed with the job's spooled output.

***OUTFILE:** The output is directed to a database file.

OUTFILE

Specifies the name and library of the database file to which the output of this command is directed. If the file does not exist, the system creates a file in the specified library. If a new file is created, the system uses file QAOYDSPCAL in system library QOFC with a record format name of DSPCAL as a model. If the file already exists, it must have this format.

The possible library values are:

***LIBL:** The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file is created in the QGPL library.

***CURLIB:** The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the file is located.

database-file-name: Specify the name of the output file.

OUTMBR

Specifies the name of the database file member that receives the output of this command.

Element 1: Member to Receive Output

***FIRST:** The first member in the file receives the output. If it does not exist, the system creates a member with the same name as the file specified on the OUTFILE parameter.

member-name: Specify the name of the file member that receives the output. If it does not exist, the system creates it.

Element 2: Operation to Perform on Member

***REPLACE:** The output data replaces existing records in the specified member.

***ADD:** The output data is added at the end of existing records in the specified member.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values. This parameter applies to the CAL parameter and the GRPCALS parameter.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEVCD:** The character set and code page defined for the device is used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples

Example 1: Displaying Scheduling Calendar Information

```
DSPCALD
```

This command displays the calendar information for the scheduling calendar of the user issuing the command.

Example 2: Printing Calendar Information

```
DSPCALD CAL(*SCHED PJC) OUTPUT(*PRINT)
```

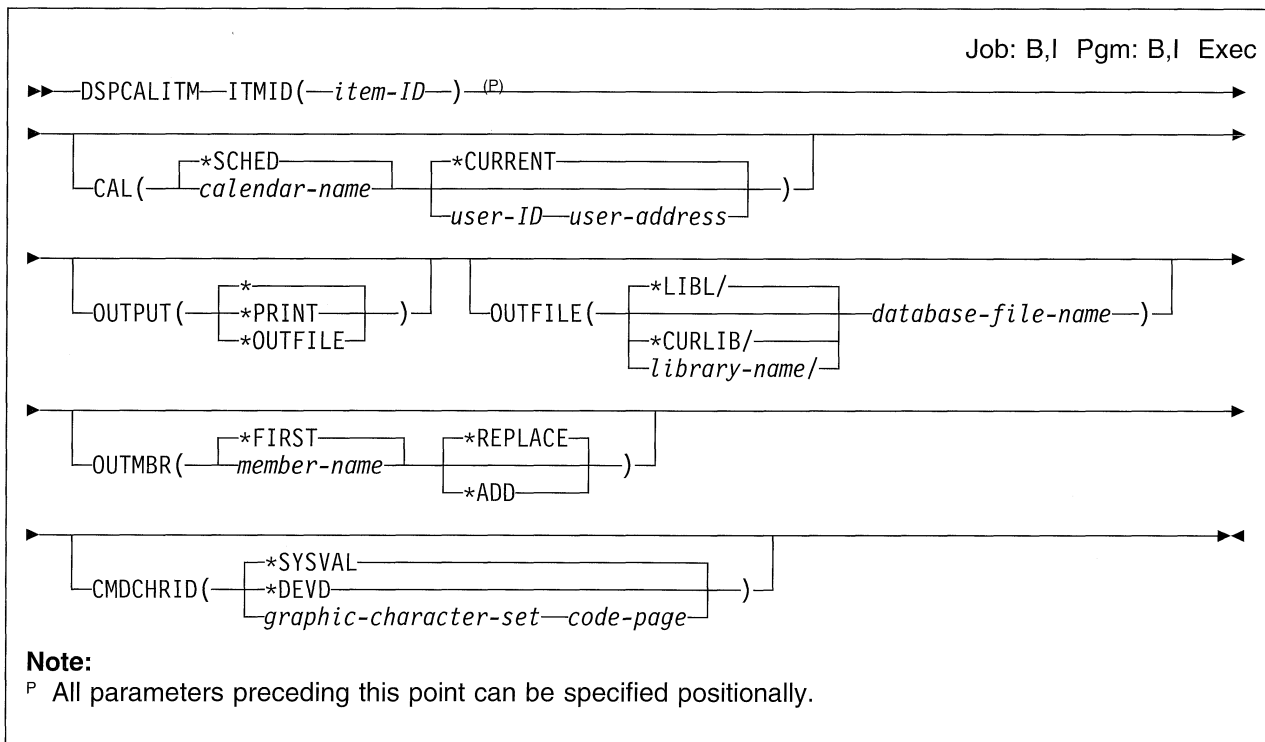
This command prints the calendar information for the scheduling calendar owned by PJC.

Example 3: Generating a Database File

```
DSPCALD CAL(CONFRM1 SEC44B RCHAS1)
OUTPUT(*OUTFILE) OUTFILE(RMSLIB/CAL)
OUTMBR(*FIRST *REPLACE)
```

This command generates a database file containing the calendar information for calendar CONFRM1.

DSPCALITM (Display Calendar Item) Command



Purpose

The Display Calendar Item (DSPCALITM) command is used to display information about an item on a local calendar. The information can be shown on a display, printed, or directed to a database file.

Note: If the user has *TIMES authority to the security level of the item, the calendar name, owner, item ID, item type, date, and time are shown. All other information is left blank.

Parameters

ITMID

Specifies the item identifier of the item to be displayed. The item identifier is generated by the system and is obtained from the Add Calendar Item (ADDCALITM) or Query Calendar Item (QRYCALITM) command. The item identifier can be a maximum of 20 characters.

CAL

Specifies the name and owner of the calendar on which the item occurs.

Element 1: Calendar Name

***SCH**ED: The item occurs on the scheduling calendar owned by the user specified on Element 2.

calendar-name: Specify the name of the calendar on which the item occurs.

Element 2: Calendar Owner

***C**RRENT: The item occurs on the calendar owned by the user issuing the command.

user-ID: Specify the user ID of the owner of the calendar on which the item occurs. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar on which the item occurs. This value is valid only if user ID is also specified.

OUTPUT

Specifies whether the output from this command is displayed, printed, or directed to a database file.

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

***PRINT:** The output is printed with the job's spooled output.

***OUTFILE:** The output is directed to a database file.

OUTFILE

Specifies the name and library of the database file to which the output is directed. If the file does not exist, the system creates a file in the specified library. If a new file is created, the system uses file QAOYDSPITM in system library QOFC with a record format name of DSPITM as a model. If the file already exists, it must have this format.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the file is located.

database-file-name: Specify the name of the output file.

OUTMBR

Specifies the name of the database file member that receives the output of this command.

Element 1: Member to Receive Output

***FIRST:** The first member in the file receives the output. If it does not exist, the system creates a member with the same name as the file specified on the OUTFILE parameter.

member-name: Specify the name of the file member that receives the output. If it does not exist, the system creates it.

Element 2: Operation to Perform on Member

***REPLACE:** The output data replaces existing records in the specified member.

***ADD:** The output data is added at the end of existing records in the specified member.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device is used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples**Example 1: Displaying an Item**

```
DSPCALITM ITMID(&ITEM)
```

This command displays the calendar item that is contained in the CL variable &ITEM from the scheduling calendar of the user issuing the command.

Example 2: Printing an Item

```
DSPCALITM ITMID(&ITEM) CAL(*SCHED PJC)
OUTPUT(*PRINT)
```

This command prints the calendar item that is contained in the CL variable ITEM from the scheduling calendar of PJC.

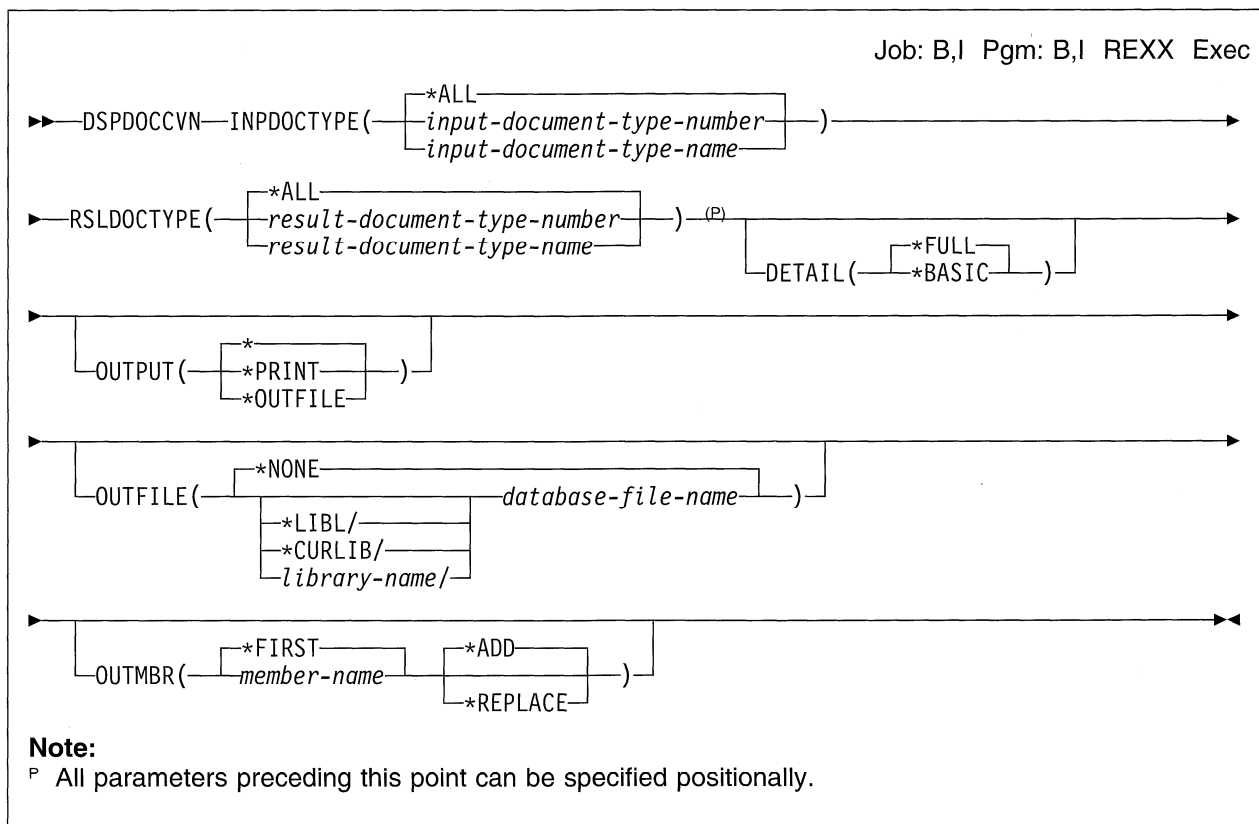
Example 3: Generating a Database File

```
DSPCALITM ITMID(&ITEM)
CAL(CONFRM1 SEC44B RCHAS1)
OUTPUT(OUTFILE) OUTFILE(RMSLIB/CAL)
OUTMBR(*FIRST *REPLACE)
```

DSPCALITM

This command generates a database file containing the calendar item in the CL variable ITEM from calendar CONFRM1.

DSPDOCCVN (Display Document Conversion) Command



Purpose

The Display Document Conversion (DSPDOCCVN) command shows a document conversion definition. The output can be directed to a screen, printer, or a database file.

Restriction: You must have security officer (*SECOFR) authority or be enrolled in OfficeVision/400 as an administrator to use this command.

Parameters

INPDOCTYPE

Specifies the name or number of the document type used as input for the document conversion definition being shown.

***ALL:** All input document types are shown.

input-document-type-number: Specify the numeric value associated with the Document Interchange Architecture (DIA) document type

requested. Valid values range from 1 through 65535.

input-document-type-name: Specify the 10-character name associated with the defined document type.

RSLDOCTYPE

Specifies the name or number of the document type used as the result type for the document conversion definition being shown.

***ALL:** All result document types are shown.

result-document-type-number: Specify the numeric value associated with the DIA document type requested. Valid values range from 1 through 65535.

result-document-type-name: Specify the 10-character name associated with the defined document type.

DETAIL

Specifies the set of attributes shown for each document conversion.

***FULL:** The output contains the name and a full set of attributes for each document conversion.

***BASIC:** The output contains the name and a basic set of attributes for each document conversion.

OUTPUT

Specifies whether the output from the command is shown at the requesting work station, printed with the job's spooled output, or directed to a database file.

***:** Output requested by an interactive job is shown on the display. Output requested by a batch job is printed with the job's spooled output.

***PRINT:** The output is printed with the job's spooled output.

***OUTFILE:** The output is directed to the database file specified on the OUTFILE parameter. All data is written to the output file regardless of the value specified on the DETAIL parameter.

OUTFILE

Specifies the qualified name of the database file to which the output of the display is directed. If the file does not exist, it is created. If the file is created, the text is "OUTFILE for DSPDOCCVN" and the public authority is *EXCLUDE.

***NONE:** The data is not written to a database file. This value must be specified for output to a display or printer.

The possible library values are:

***LIBL:** The library list is used to locate the calendar.

***CURLIB:** The current library for the job is used to locate the calendar. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the calendar is located.

database-file-name: Specify the name of the database file that receives the output of the display.

OUTMBR

Specifies the name of the database file member to which the output of the display is directed. If a member already exists, and *REPLACE is specified, the system clears the member and adds the new records. If the member does not exist and a member name is not specified, the system creates a member in the file specified on the OUTFILE parameter.

Element 1: Member to Receive Output

***FIRST:** The first member in the file receives the output.

member-name: Specify the name of the file member to receive the output.

Element 2: Operation to Perform on Member

***ADD:** If a member exists, the system adds the new records to the end of the existing records.

***REPLACE:** If a member exists, the system clears it and adds the new records.

Examples

Example 1: Displaying Conversion Details

```
DSPDOCCVN INPDOCTYPE(RFTDCA) RSLDOCTYPE(RFTAS400)
```

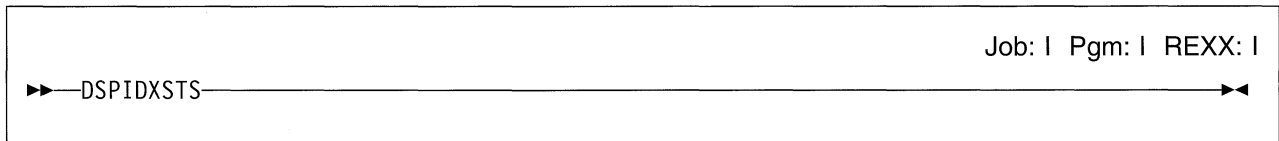
This command shows all the details of the specified conversion.

Example 2: Printing Conversion Details

```
DSPDOCCVN INPDOCTYPE(*ALL) RSLDOCTYPE(*ALL)
OUTPUT(*PRINT) DETAIL(*BASIC)
```

This command prints all the conversions defined to the system in a list format.

DSPIDXSTS (Display Text Index Status) Command



Purpose

The Display Text Index Status (DSPIDXSTS) command shows information that indicates when documents are scheduled to be indexed and when they have been indexed. The Display Text Index Status display shows the following information:

- Whether the text index can be searched or updated
- The number of documents on the scheduling queue
- Whether a text index update is currently running

- The date and time of the previous text index update
- The date and time of the next text index update

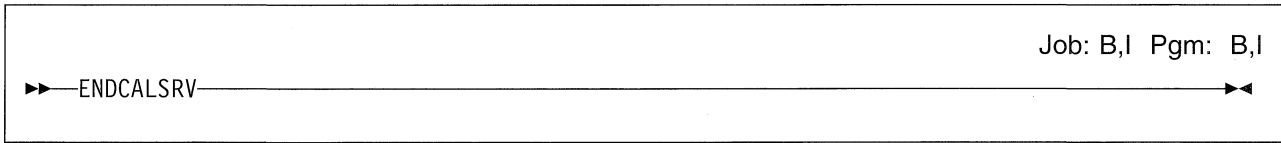
There are no parameters for this command.

Example

```
DSPIDXSTS
```

This command shows information that indicates the index status of documents.

ENDCALSRV (End Calendar Service) Command



Purpose

The End Calendar Service (ENDCALSRV) command ends the current calendar service job. The job is ended in a controlled manner with a delay time of 30 seconds for cleanup.

There are no parameters for this command.

Example

ENDCALSRV

This command ends the current calendar service job.

ENDIDXMN (End Index Monitor) Command

| |
|----------------------------------|
| Job: B,I Pgm: B,I REXX: B,I Exec |
| ▶▶—ENDIDXMN—◀◀ |

Purpose

The End Index Monitor (ENDIDXMN) command requests that the index monitor be stopped at the next polling cycle.

Update or reorganization operations that are already in progress are not interrupted.

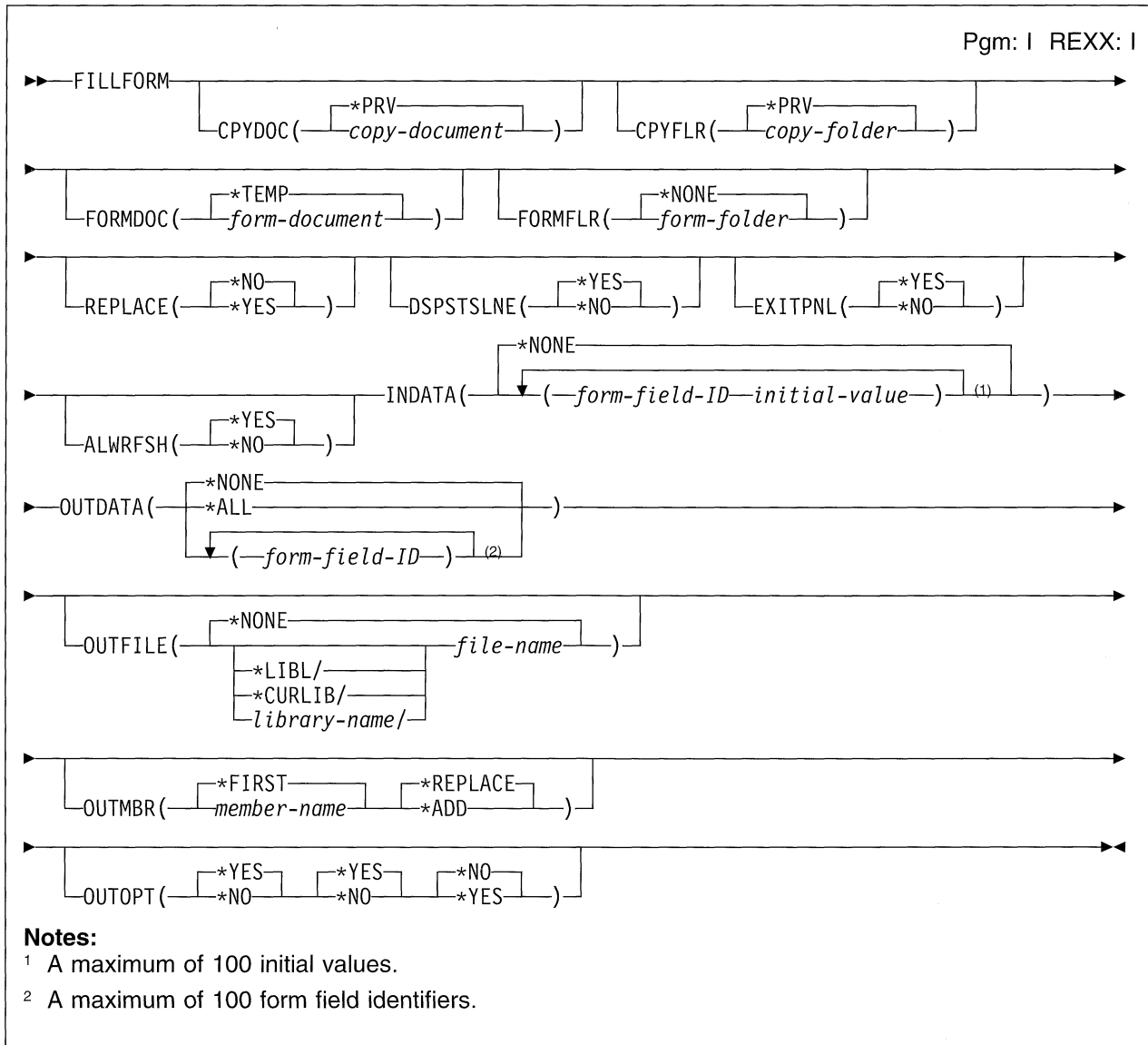
There are no parameters for this command.

Example

```
ENDIDXMN
```

This command stops the index monitor.

FILLFORM (Fill Form Document) Command



Purpose

The Fill Form Document (FILLFORM) command allows the user to complete a form using the fill form editor of OfficeVision/400.

Parameters

CPYDOC

Specifies the name of the shell document that is used to create the form document.

***PRV:** The name of the document used in your last session.

copy-document: Specify the name of the shell document that is used to create the form document.

CPYFLR

Specifies the name of the folder containing the shell document.

***PRV:** The name of the folder that was used in the last session.

copy-folder: Specify the name of the folder that contains the shell document.

FORMDOC

Specifies the name of the document where the current form is saved when the fill form session ends.

***TEMP:** The current form document is temporary and is not saved when the fill form session ends.

form-document: Specify the name of the document where the current form is saved.

Note: If FORMDOC is specified, the EXITPNL(*NO) or ALWRFSH(*NO) must also be specified.

FORMFLR

Specifies the name of the folder where the current form is saved.

***NONE:** The current form document is not saved.

form-folder: Specify the name of the folder where the current form is saved.

REPLACE

Specifies whether to replace an existing form document.

***NO:** Do not replace the form document.

***YES:** Replace the form document.

DSPSTSLNE

Specifies whether the fill form edit status lines 1 and 2 will be shown on the the fill form display.

***YES:** Status lines 1 and 2 will be on the display.

***NO:** Status lines 1 and 2 will not be on the display.

EXITPNL

Specifies whether the Exit Form Document Display should be shown when F3(Exit) or F12(Cancel) ends the editing.

***YES:** The Exit Form Document display appears when F3 or F12 is pressed

***NO:** The Exit Form Document display does not appear when F3 or F12 is pressed.

ALWRFSH

Specifies whether the form refresh function (F5) will be allowed from within the editor and on the exit panel.

***YES:** The refresh option will be allowed.

***NO:** The refresh option will not be allowed.

INDATA

Specifies a list of form fields with values that will override the initial values stored in the document. A maximum of 100 initial values can be used.

***NONE:** No form fields will be overridden.

Element 1: form-field-ID

Specifies the form field identifier(s) that will be overridden.

form-field-ID: A form field identifier that specifies a form field to be overridden.

Element 2:'initial value'

Specifies an initial value for the form field ID specified. This value overrides any initial value stored in the FORMDOC.

OUTDATA

Specifies a list of form fields whose contents will be written to the database file specified in the OUTFILE parameter. A maximum of 100 form field identifiers can be used.

Element 1: form-field-ID

Specifies the form field identifier(s) to be written. The possible values are:

***NONE:** No form fields will be written.

***ALL:** All form fields with form field identifiers will be written.

form-field-ID: A form field identifier that specifies a form field to be written.

OUTFILE

Specifies the name and library of a database file in which to write the current contents of the form field data. The specifications of the OUTDATA parameter will determine how the data will be written.

***NONE:** No form field data will be written to the database.

***LIBL:** The library list of the job will be used to search for the database file. If the file is not found, one is created in the current library. If no current library exists, the file is created in the QGPL library.

***CURLIB:** The current library of the job will be used to search for the database

FILLFORM

file. If no library is specified as the current library for the job, QGPL is used.

library-name: Specify the name of the library that contains the database file.

file-name: Specify the name of the database file used for output.

OUTMBR

Specifies the name of the database file member in which to write the current contents of the form field data.

Element 1: member-name

Specifies the name of the database file member used for output. The possible values are:

***FIRST:** The database file's first member will be used for output.

member-name: Specifies the name of the database file member for output.

Element 2: replace

Specifies if the data should be replaced. The possible values are:

***REPLACE:** The data in the specified member will be replaced. If the member specified does not exist, it will be created.

***ADD:** The data will be added to the end of the existing member. If the member does not exist, it will be created.

OUTOPT

Specifies when data will be written to the specified output file member.

Element 1: 'output data on exit'

Specifies if data will be written when exiting the fill form session. The possible values are:

***YES:** The data will be written when exiting a fill form session.

***NO:** The data will not be written when exiting a fill form session.

Element 2: 'output data on refresh'

Specifies if data will be written when refreshing a form. The possible values are:

***YES:** The data will be written when refreshing a form.

***NO:** The data will not be written when refreshing a form.

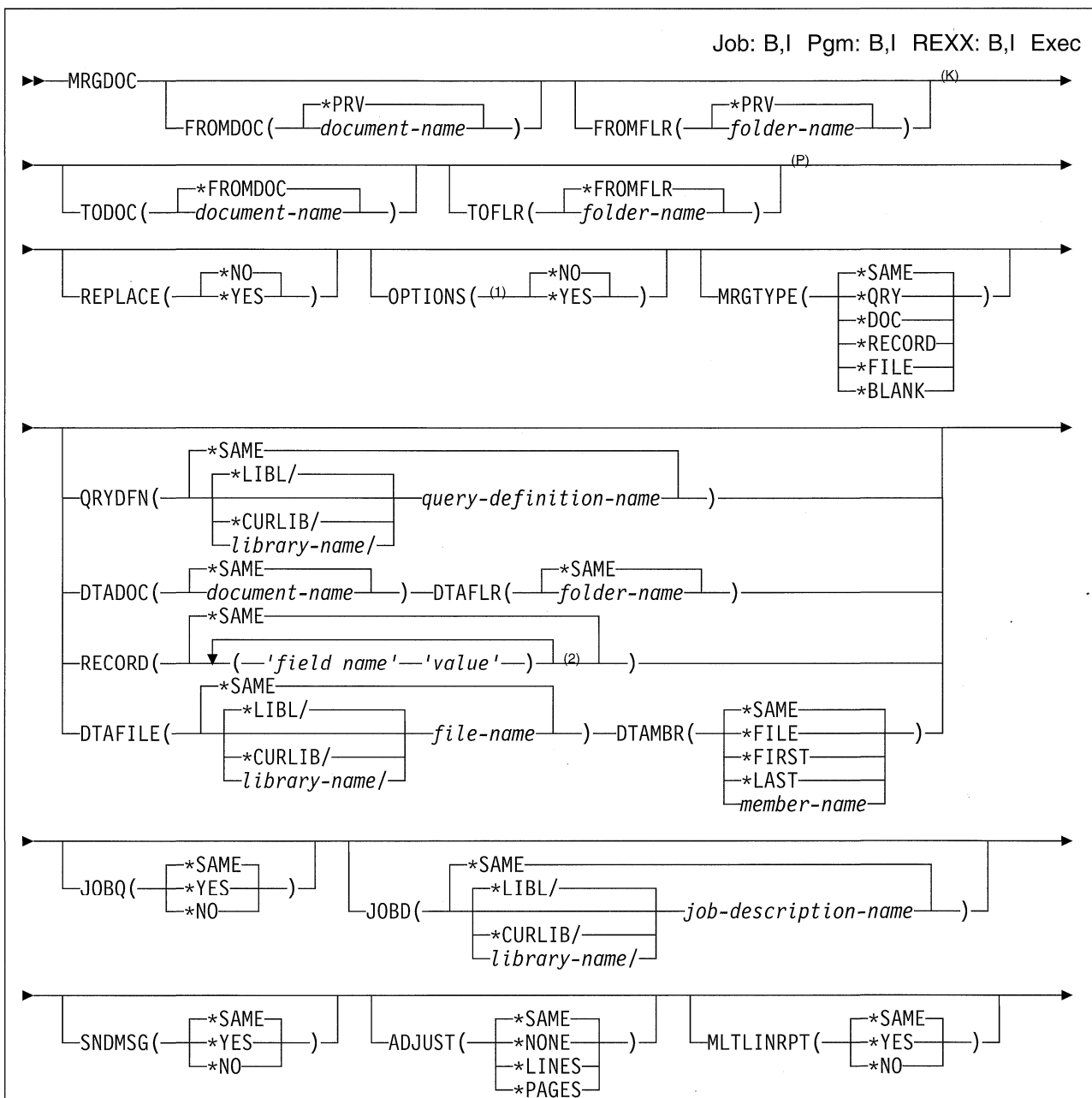
Element 3: 'output data when errors exist'

Specifies if data will be written when any field in the form is in error. The possible values are:

***NO:** The data will not be written when errors exist in the data.

***YES:** The data will be written when errors exist in the data.

MRGDOC (Merge Document) Command



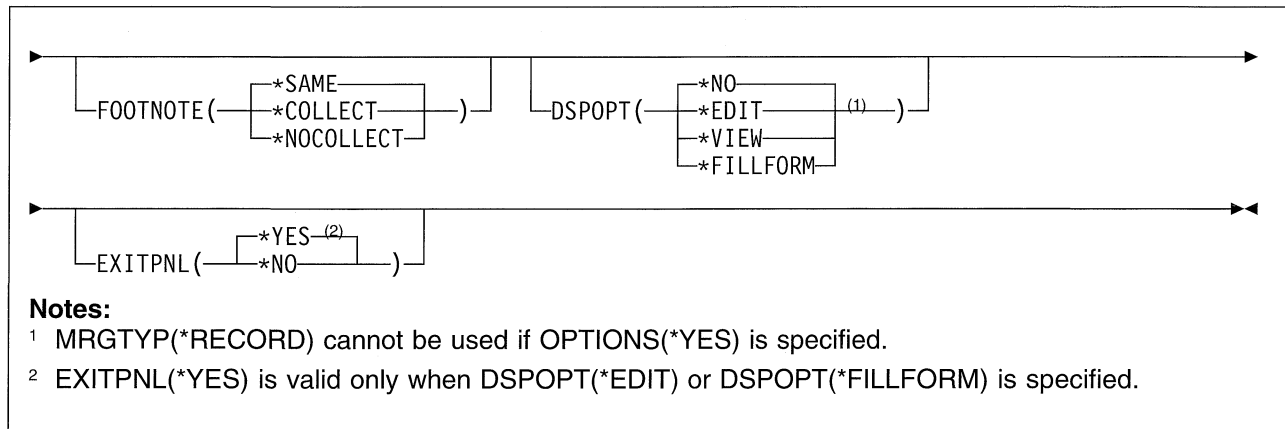
Notes:

^K All parameters preceding this point are key parameters.

^P All parameters preceding this point can be specified positionally.

¹ Batch cannot be used if OPTIONS(*YES) is specified.

² A maximum of 50 repetitions.



Purpose

The Merge Document (MRGDOC) command merges text and data into a document that can be edited by using the word processing function of OfficeVision/400.

When a document is created, a set of default merge options are associated with that document. Users can override one or more of the default merge options without using the merge options display by specifying the appropriate parameters on this command.

MRGDOC processes the following text-coding instructions:

- .& (data field)
- .dfh (data field heading)
- .inc (include)
- .bct (begin conditional text)
- .ect (end conditional text)
- .set (set variable)

More information on merging documents is in the *Using OfficeVision/400* Word Processing* manual.

Parameters

FROMDOC

Specifies the name of the document being merged with the current document.

***PRV:** The name of the document used in the previous session is merged with the current document.

document-name: Specify the name of the document being merged with the current document.

FROMFLR

Specifies the name of the folder that contains the document being merged with the current document.

***PRV:** The name of the folder used in the last session is used.

folder-name: Specify the name of the folder that contains the document being merged with the current document.

TODOC

Specifies the name of the output document for the merged information.

***FROMDOC:** The name of the document from which the user is merging information is used.

document-name: Specify the name of the output document for the merged information.

TOFLR

Specifies the name of the folder that contains the output document for the merged information.

***FROMFLR:** The name of the folder from which the user is merging information is used.

folder-name: Specify the name of the folder that receives the output document for the merged information.

REPLACE

Specifies whether an existing output document is replaced by the merge operation.

***NO:** If the specified TODOC exists, it is not replaced and the merge is not performed.

***YES:** If the specified TODOC exists, it is replaced with the resulting document from the merge operation.

OPTIONS

Specifies whether the merge options for this document are changed before the document is merged.

***NO:** The merge options are not displayed and changed.

***YES:** The merge options are displayed and changed before the document is merged.

Note: To change the merge options, you must have change authority to the document.

MRGTYPE

Specifies the type and location of the data being merged. This parameter provides the capability to specify the type of data being merged, whether it is data from a query, data stored in a file, or data stored in a document.

Note: Identify all Data Field and Begin Conditional Text instructions for text by specifying option 2 (*PRINT) for the Data Field source prompt.

***SAME:** The type of data specified in the document merge options does not change.

***QRY:** The data is merged from a query source. This query is a request to select and copy data from a file of one or more records based on the defined conditions.

***DOC:** The data stored in a document is merged.

***RECORD:** The data to be merged is passed directly on the MRGDOC API call. The data comprises one or more fields of a single record. The data used must be character data.

***FILE:** The data stored in a file is merged.

***BLANK:** No data is merged.

The following six parameters are mutually exclusive.

QRYDFN

Specifies the name of the query that defines the data being merged.

***SAME:** The query name specified in the document merge options does not change.

***LIBL:** The library list is used to locate the query name.

***CURLIB:** The current library for the job is used to locate the query name. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the query name is located.

query-definition-name: Specify the name of the query used to move the merged data.

DTADOC

Specifies the name of the document that contains the data being merged. This parameter is valid only when MRGTYPE(*DOC) is selected.

***SAME:** The document name specified in the document merge options does not change.

document-name: Specify the name of the document using 1 to 12 alphanumeric characters. If more than 8 characters are used, the ninth character must be a period (.) followed by a 1 to 3 character extension.

DTAFLR

Specifies the name of the folder containing the document being merged. This parameter is valid only if MRGTYPE(*DOC) is selected.

***SAME:** The folder name specified in the document merge options does not change.

folder-name: Specify the name of the folder containing the document being merged.

RECORD

Specifies up to 50 'field-name/value' combinations. This parameter is valid only if MRGTYPE(*RECORD) is specified. The field name can be a data field name from the database record and the value can be a CL program variable.

DTAFILE

Specifies the file that contains the data being merged. This parameter is valid only when MRGTYPE(*FILE) is selected.

***SAME:** The file name specified in the document merge options does not change.

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the file is located.

file-name: Specify the name of the file that contains the data to be moved.

DTAMBR

Specifies the member that contains the data being merged. This parameter is valid only when MRGTYPE(*FILE) is selected.

***SAME:** The member name specified in the document merge options does not change.

***FILE:** The member with the same name as the file name is used.

***FIRST:** The first member is used.

***LAST:** The last member is used.

member-name: Specify the name of the member that contains the data being merged.

JOBQ

Specifies whether the merge request is put on the job queue.

***SAME:** The place on the job queue that is specified in the document merge options does not change.

***YES:** The merge request is placed on the job queue.

***NO:** The merge request is not placed on the job queue.

JOB D

Specifies the name of the job description that describes how the merge job is run.

***SAME:** The name of the job description that is specified in the document merge options does not change.

***LIBL:** The library list is used to locate the job description.

***CURLIB:** The current library for the job is used to locate the job description. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the job description is located.

job-description-name: Specify the name of the job description.

SN DMSG

Specifies the merging of a job to the job queue and that the user wants to receive a message when the job is completed.

***SAME:** The message value specified in the document merge options does not change.

***YES:** A message is sent when the merge has completed.

***NO:** A message is not sent when the merge has completed.

ADJUST

Specifies that line endings and page endings in the merged document are adjusted.

***SAME:** The adjust value specified in the document merge options does not change.

***NONE:** Line endings and page endings are not adjusted in the merged document.

***LINES:** Only the line endings in the merged document are adjusted.

***PAGES:** The line endings and the page endings in the merged document are adjusted.

MLTLINRPT

Specifies a multiple line report. In this report, data field instructions are merged to create records with several lines of output.

***SAME:** The multiple-line-report option specified in the document merge options is used.

***YES:** A multiple line report is created.

***NO:** A multiple line report is not created.

FOOTNOTE

Specifies that all footnote text, associated with the footnote instructions found in the document, is collected in the merged document.

***SAME:** The collect footnote value specified in the document merge options does not change.

***COLLECT:** The footnote text and the associated footnote instructions are collected and become part of the merged document.

***NOCOLLECT:** The footnote text and the associated footnote instructions are not collected.

DSPOPT

Specifies whether the document is edited or viewed in the same job step as the merge operation.

***NO:** The document is neither edited nor viewed during the merge operation.

***EDIT:** The document is edited during the merge operation.

***VIEW:** The document is viewed during the merge operation.

***FILLFORM:** The document is edited during the merge operation. The fill form editor will be invoked against the merged document.

EXITPNL

Specifies whether the Exit Document display is shown when F3(Exit) or F12(Cancel) is pressed to end the editing.

***YES:** The Exit Document display is shown when F3(Exit) or F12(Cancel) is pressed to end the editing.

***NO:** The Exit Document display is not shown when F3(Exit) or F12(Cancel) is pressed to end the editing.

Examples**Example 1: Showing Merge Options Before Merge Operation**

```
MRGDOC FROMDOC(YOURDOC)
      FROMFLR(YOURFLR) TODOC(MYDOC)
      TOFLR(MYFLR) REPLACE(*YES)
      OPTIONS(*YES)
```

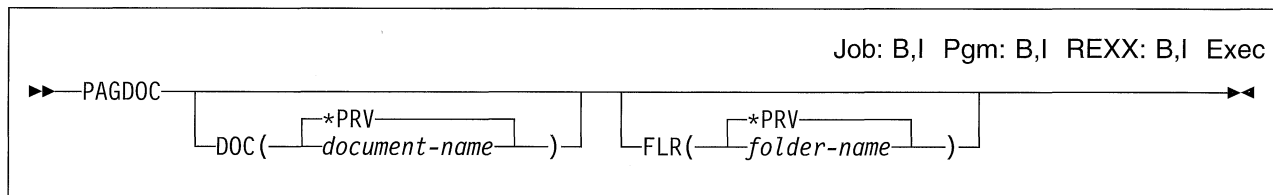
This command merges document YOURDOC in folder YOURFLR and places the output in document MYDOC in folder MYFLR. The merge options are shown before the merge takes place. If a document already exists by the name of MYDOC, it is replaced by the newly merged document.

Example 2: Merging a Document From a Fill-In Document and Folder

```
MRGDOC FROMDOC(YOURDOC)
      FROMFLR(YOURFLR) TODOC(MYDOC)
      TOFLR(MYFLR) OPTIONS(*NO)
      MRGTYPE(*DOC) DTADOC(TEXTINFO)
      DTAFLR(TEXTINFO)
```

This command merges a document from a fill-in document and folder called TEXTINFO.

PAGDOC (Paginate Document) Command



Purpose

The Paginate Document (PAGDOC) command paginates a document using the word processor function of OfficeVision/400. More information on paginating documents is in the *Using OfficeVision/400* Word Processing* manual.

Parameters

DOC

Specifies the name of the document being paginated.

***PRV:** The name of the document used in the previous session is paginated.

document-name: Specify the name of the document being paginated.

FLR

Specifies the name of the folder that contains the document.

***PRV:** The name of the folder used in the previous session is used.

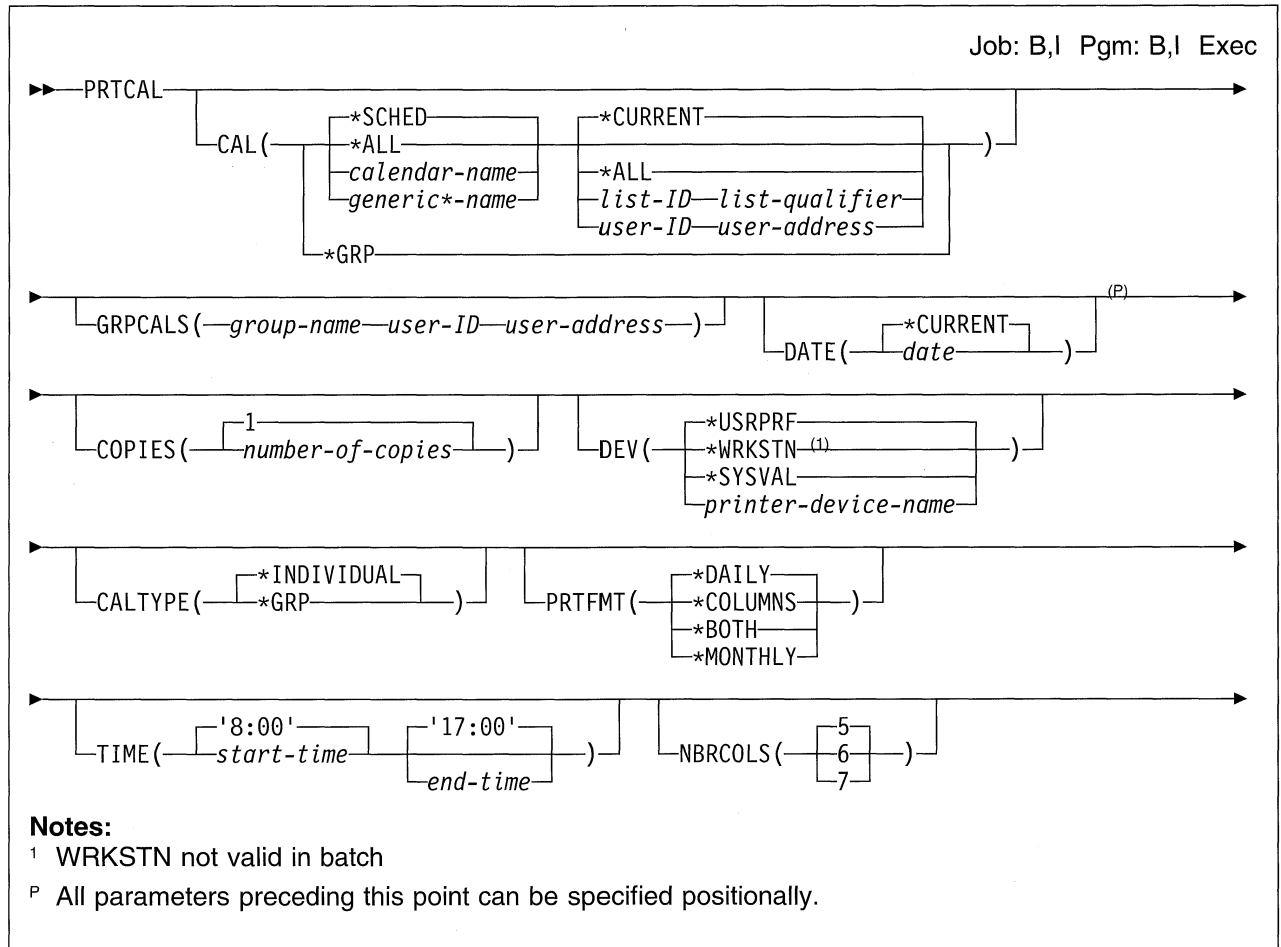
folder-name: Specify the name of the folder that contains the document to paginate.

Example

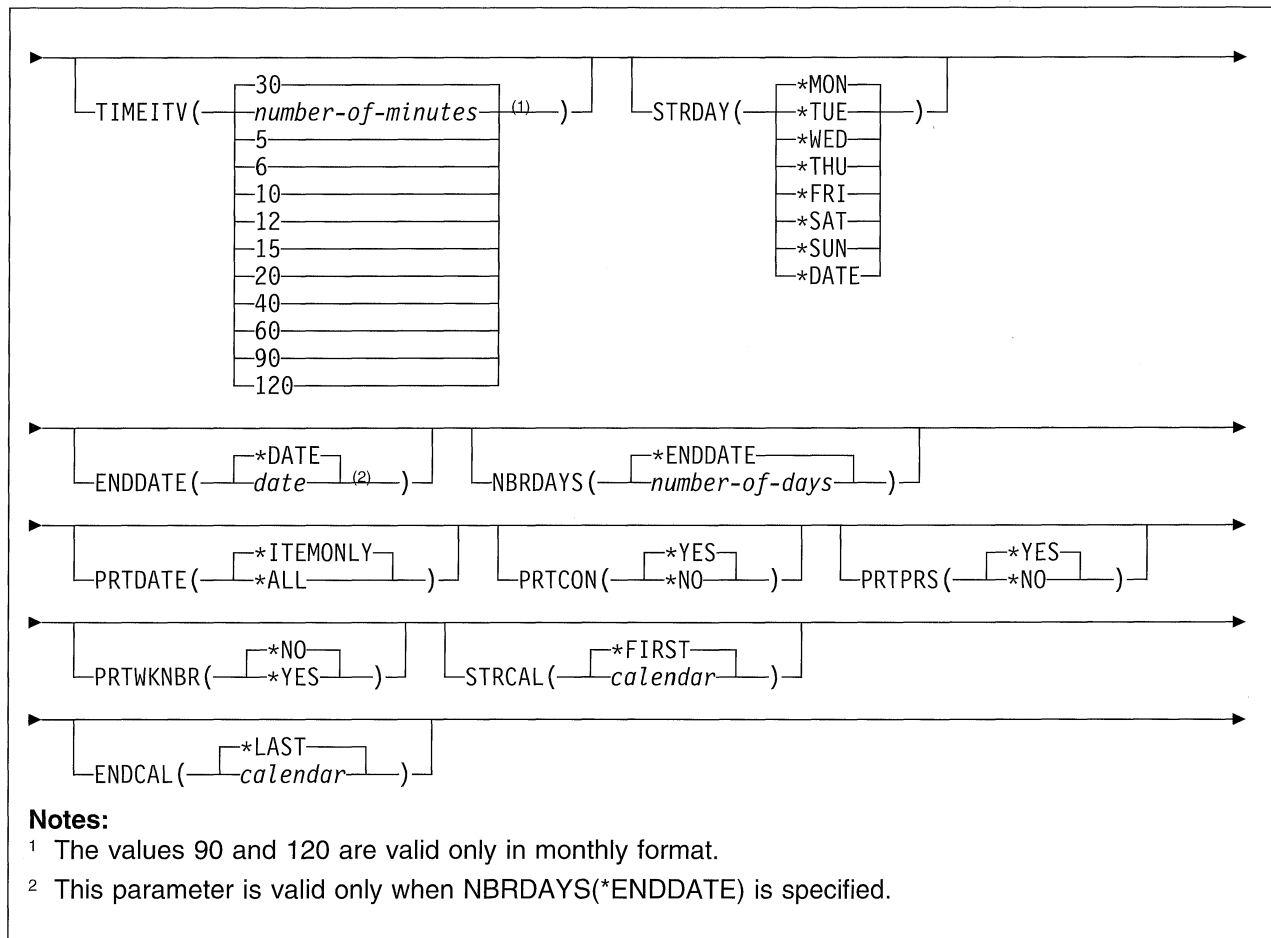
```
PAGDOC DOC(VDOC) FLR(MYFLR)
```

This command paginates document VDOC in folder MYFLR.

PRTCAL (Print Calendar) Command

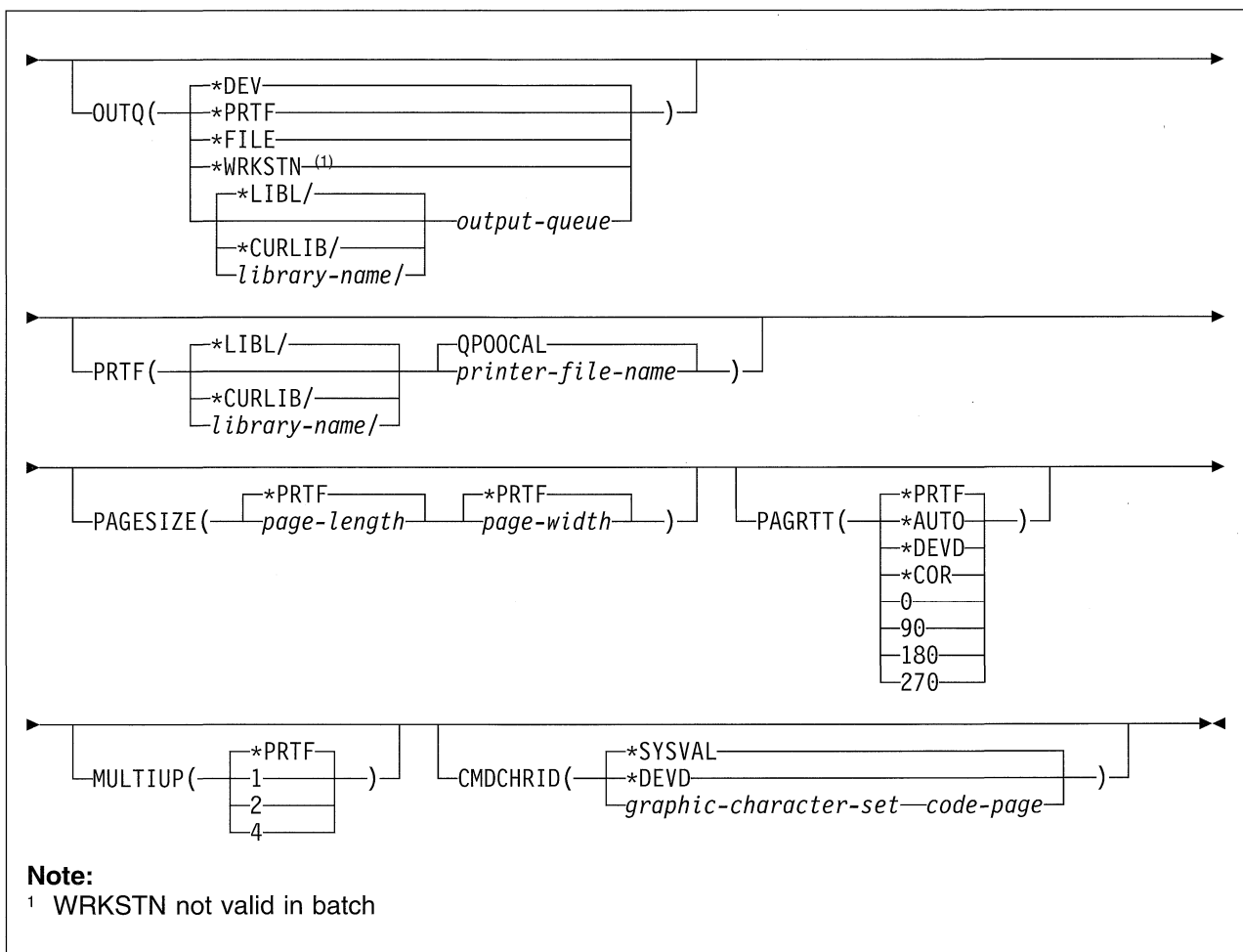


PRTCAL



Notes:

- ¹ The values 90 and 120 are valid only in monthly format.
- ² This parameter is valid only when NBRDAYS(*ENDDATE) is specified.



Purpose

The Print Calendar (PRTCAL) command is used to print one or more calendars. Calendars can be printed in a weekly column format, in a daily format, both formats, or a monthly format. Groups of calendars and monthly calendars also can be printed in a column, list, or monthly format. If the user is enrolled in OfficeVision/400, the format of the calendar printout defaults to the current enrollment values. Any values specified on this command override the enrollment values. If the user is not enrolled in OfficeVision/400, the default values for this command are used.

Parameters

CAL

Specifies the name and owner of the calendar to be printed.

Note: The user must have at least *TIMES authority to the calendar to print the calendar.

Element 1: Calendar Name

***SCHED:** The scheduling calendars owned by the users specified on the second element of this parameter are printed.

***ALL:** All calendars owned by the users specified on the second element of this parameter are printed.

calendar-name: Specify the name of the calendar to be printed.

generic-name:* Specify the generic name of the calendar to be printed. A generic name is a character string that contains one or more characters followed by an asterisk (*). Calendars that have names that match the generic name and are owned by the users specified on the second element of this parameter are printed.

Note: Any calendar name ending in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify 'MIKE**' to print all calendars with names beginning with 'MIKE*'.
Element 2: User Information

***CURRENT:** The calendar owned by the user issuing the command is printed.

***ALL:** All calendars that have the name specified on the first element of this parameter and are owned by local users are printed.

list-identification: Specify a distribution list. All calendars that have the name specified on the first element of this parameter and that are owned by users on the distribution list are printed.

user-ID: Specify the user ID of the owner of the calendar to be printed. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar to be printed. This value is valid only if user ID is also specified.

Other Single Values:

***GRP:** All of the calendars in the calendar group specified on the GRPCALS parameter are printed.

Note: If *GRP is specified, the GRPCALS parameter must also be specified..

GRPCALS

Specifies the local calendar group to be printed. This parameter is valid only if CAL(*GRP) is also specified. A value for Element 1 must be specified if GRPCALS is specified.

Element 1: Group Name

group-name: Specify the name of the calendar group to be printed.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group.

Element 3: User Address

address: Specify the user address of the owner of the calendar group.

DATE

Specifies the start date. All items on the calendar that occur on or after the date specified are printed.

***CURRENT:** All items that occur on or after the current date are printed.

date: Specify the start date, which must be specified in the job date format.

COPIES

Specifies the number of copies to be printed.

1: One copy of the calendar is printed.

number-of-copies: Specify the number of copies of the calendar to print. Valid values range from 1 through 9999.

DEV

Specifies the name of the printer used to print the calendar.

***USRPRF:** The printer specified in the user profile of the user issuing the command is used.

***WRKSTN:** The printer specified on the PRTDEV parameter of the device description is used. This value is not valid if the command is run in batch.

***SYSVAL:** The printer specified on the QPRTDEV system value is used.

printer-device-name: Specify the name of the printer.

CALTYP

Specifies the type of view shown for each calendar printed.

***INDIVIDUAL:** The individual view of the calendars is printed.

***GRP:** The group view of the calendars is printed.

PRTFMT

Specifies the format of the calendars printed.

***DAILY:** The calendar is printed in a format similar to the daily view. The printout shows a list of items that occur on the calendar. This value is not allowed if *GRP is specified on the CALTYP parameter.

***COLUMNS:** The calendar is printed in a column format. If *INDIVIDUAL is specified on the CALTYP parameter, each column represents a day. If *GRP is specified on the CALTYP parameter, each column represents a calendar.

***BOTH:** The calendar is printed in column format, followed by the daily format.

***MONTHLY:** The calendar is printed in monthly format.

TIME

Specifies the time range that is used with the column format or the starting time that is used with the monthly format.

For the column format, each column begins with the time specified on the first element of this parameter and ends with the time specified on the second element.

For the monthly format, each date cell begins with the time specified on the first element of this parameter and the second element is ignored. The system calculates the end time from the Start Time, the NBRCOLS, and the TIMEITV.

Note: This parameter is not valid if *DAILY is specified on the PRTFMT parameter.

Element 1: Start Time

0800: The columns start at 8 a.m.

start-time: Specify the time the cell or columns start.

Element 2: End Time

1700: The columns end at 5 p.m.

end-time: Specify the time the cell or columns end.

NBRCOLS

Specifies the number of columns that are printed. If fewer columns are specified, more calendar item information appears on the printout.

Note: This parameter is not valid if *DAILY is specified on the PRTFMT parameter.

5: Five columns are printed.

6: Six columns are printed.

7: Seven columns are printed.

TIMEITV

Specifies how much time each line of the column format printout or each point in a date cell of the monthly format represents. For the column format, the fewer number of minutes specified, the more calendar item information is printed.

Note: This parameter is not valid if *DAILY is specified on the PRTFMT parameter.

For column format, the number of lines on the printout cannot exceed 144 lines. To determine how many lines are taken up by the number of minutes specified on this parameter, use the following: (number of hours on the calendar)*(60/TIMEITV)

30: Each line or point of the printout represents 30 minutes.

number-of-minutes: Specify the number of minutes each line represents. When printing a calendar in a weekly format a user can specify 5, 6, 10, 12, 15, 20, 40, or 60. When printing a calendar in a monthly format, a user can specify 5, 6, 10, 12, 15, 20, 30, 40, 60, 90, or 120.

Note: A value greater than 60 is not valid for this parameter if *WEEKLY or *BOTH is specified on the PRTFMT parameter.

STRDAY

Specifies the day where the left-most column will be on a weekly or a monthly calendar output.

Note: This parameter is not valid if *DAILY is specified on the PRTFMT parameter. The default value will be ignored if *DAILY is specified.

***MON:** The start day is a Monday.

***TUE:** The start day is a Tuesday.

***WED:** The start day is a Wednesday.

***THU:** The start day is a Thursday.

***FRI:** The start day is a Friday.

***SAT:** The start day is a Saturday.

***SUN:** The start day is a Sunday.

***DATE:** The start day is the day of the DATE parameter.

PRTCAL

ENDDATE

Specifies the date of the week or month on which the ENDDATE falls. The printout is stopped on the ENDDATE.

Note: The printout of a weekly or monthly calendar will always be full weeks or months no matter where the ENDDATE falls during the week or month. This parameter is valid for all options on the PRTFMT parameter.

***DATE:** The end date is the same as the date specified on the DATE parameter. Only one day's calendar items are printed.

date: Specify the date in the job date format.

NBRDAYS

Specifies the number of days printed in the print job.

***ENDDATE:** The value specified on the ENDDATE parameter determines the number of days included in the print job.

number-of-days: Specify the number of days to include in the print job. Valid values range from 1 through 9999.

PRTDATE

Specifies whether all dates between the start and end dates are printed on the daily print job, or only the dates that contain at least one item.

***ITEMONLY:** The dates that have at least one item are printed.

***ALL:** All dates are printed.

PRTCON

Specifies whether confidential items are printed.

***YES:** Confidential items are printed if the user is authorized to view confidential material.

***NO:** Confidential items are not printed.

PRTPRS

Specifies whether personal items are printed.

***YES:** Personal items are printed if the user is authorized to view personal material.

***NO:** Personal items are not printed.

PRTWKNBR

Specifies whether or not to print the week number.

***NO:** Do not print the week number.

***YES:** Print the week number.

STRCAL

Specifies the name of the calendar to be printed first.

***FIRST:** The first calendar is printed first.

calendar: Specify the name of the calendar to be printed first.

ENDCAL

Specifies the name of the calendar to be printed last.

***LAST:** The last calendar is printed last.

calendar: Specify the name of the calendar to be printed last.

OUTQ

Specifies the qualified name of the output queue for the printed output.

***DEV:** The output queue specified on the DEV parameter is used.

***PRTF:** The output queue specified in the printer file on the PRTF parameter is used. This is the same as *FILE.

***FILE:** The output queue specified in the printer file on the PRTF parameter is used.

***WRKSTN:** The output queue specified on the OUTQ parameter of the Create Device Description (Display) (CRTDEV DSP) command and the Change Device Description (Display) (CHGDEV DSP) command is used.

output-queue: Specify the name of the output queue.

The possible library values are:

***LIBL:** The library list is used to locate the output queue.

***CURLIB:** The current library for the job is used to locate the output queue. If no library is specified as the current library for the job, QGPL is used.

library-name: Specify the name of the library where the output queue is located.

PRTF

Specifies the name of the printer file used when printing the calendar.

QPOOCAL: The IBM-supplied calendar printer file is used.

printer-file-name: Specify the name of the printer file.

The possible library values are:

***LIBL:** The library list is used to locate the printer file.

***CURLIB:** The current library for the job is used to locate the printer file. If no library is specified as the current library for the job, QGPL is used.

library-name: Specify the name of the library where the printer file is located.

PAGESIZE

Specifies the length and the width of the printer forms. The length is specified in lines per page. The width is specified in print positions (characters) per line. Small values are not recommended because the values will produce a large number of pages and the output will not be meaningful.

Note: The overflow line number is calculated by subtracting from the page length the difference between the page length and the overflow line number in the printer file. The result will be greater than or equal to 9.

Element 1: Page Length

***PRTF:** The value of the page length is the same as that specified in the printer file.

page-length: Specify the page length that is used. Valid values range from 15 through 255.

Element 2: Page Width

***PRTF:** The value of the page width is the same as that specified in the printer file.

page-width: Specify the page width that is used. Valid values range from 5 through 378.

PAGRRT

Specifies the degree of rotation of the calendar on the page with respect to the way the form is loaded in the printer.

***PRTF:** The degrees of page rotation is equal to that specified in the printer file. This parameter is valid only for printers supporting rotation.

***AUTO:** Indicates that automatic rotation of output is done to fit the printed data on the form. If rotation does not accomplish this, the system does it automatically. This parameter is valid only for printers that support rotation.

***DEVD:** The operating system sends a device default rotation value to the printer. Page rotation depends on the printer device specifications. To determine how your printer handles page rotation, read the documentation that came with the printer device.

***COR:** Computer output reduction is performed. This allows the user to print documents which are 13.2 inches wide by 11.0 inches long on 8.5 by 11.0 inch paper.

0: The output is not rotated. Printing starts at the edge loaded into the printer first and is parallel to that edge.

90: The output is rotated 90 degrees clockwise from the 0-degree writing position.

180: The output is rotated 180 degrees clockwise from the 0-degree writing position.

270: The output is rotated 270 degrees clockwise from the 0-degree writing position.

MULTIUP

Specifies whether or not multiple pages of output are printed on one physical page.

Note: This is supported only on IPDS printers.

***PRTF:** The number of pages of output per physical page is the equal to the value specified in the printer file.

1: One page of output is printed on each physical page.

2: Two pages of output are printed on each physical page.

4: Four pages of output are printed on each physical page.

CMDCHRID

Specifies the character identifier (graphic character set and code page) for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

PRTCAL

***SYSVAL:** The system value, QCHRID, is used for the character set and code page.

***DEV D:** The character set and code page defined for the device is used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples

Example 1: Printing a Scheduling Calendar

```
PRTCAL CAL(*SCHED) DATE(*CURRENT)
```

This command prints the scheduling calendar of the user issuing the command.

Example 2: Printing a Scheduling Calendar

```
PRTCAL CAL(*SCHED PJC *LCL) DATE(030190)  
PRTFMT(*COLUMNS) DEV(PRT01)
```

This command prints the scheduling calendar owned by local user PJC. It prints the week including March 1, 1990. It prints on printer PRT01 in weekly format.

Example 3: Printing Calendars in a Group

```
PRTCAL GRPCALS(DEPT54P) CALTYP(*GROUP)  
PRTFMT(COLUMNS)
```

This command prints the calendars listed in calendar group DEPT54P. The calendars are printed in a group format in 5 columns.

Example 4: Printing Three Copies of a Calendar

```
PRTCAL COPIES(3) PRTDATE(*ALL) DATE(122591)  
NBRDAYS(14)
```

This command prints 3 copies of the user's scheduling calendar for 14 days beginning with 12/25/91.

Example 5: Printing a Calendar Without Personal Items

```
PRTCAL CAL(*SCHED MIKE ROCH)  
DATE(100191) ENDDATE(123191)  
NBRDAYS(*ENDDATE)  
PRTDATE(*ITEMONLY) PRTPRS(*NO)
```

This command prints the scheduling calendar owned by user MIKE ROCH. Only the dates with items are shown. No personal items are printed. The start date is 10/01/91. The end date for the printout is 12/31/91.

Example 6: Printing Calendars in a Group

```
PRTCAL CAL(*GRP) GRPCALS(FINANCE)  
DATE(120191) CALTYP(*INDIVIDUAL)  
PRTFMT(*DAILY) COPIES(2) NBRDAYS(28)  
PRTDATE(*ALL) PRTCON(*NO) PRTPRS(*NO)
```

This command prints the calendars listed in group FINANCE. Two copies of each calendar are printed individually in daily format. Each calendar has date and day of the week headings for all dates between 12/01/91 and 12/28/91. Personal and confidential items are not printed.

Example 7: Printing a group calendar in column format individually

```
| PRTCAL CAL(*GRP)  
| GRPCALS(GRPCHANGE) DATE(042792)  
| ENDDATE(051292)  
| CALTYP(*INDIVIDUAL)  
| PRTFMT(*COLUMNS)
```

| This command prints the three full weeks of the first calendar in the group GRPCHANGE followed by other calendars in the group.

Example 8: Printing a group calendar in column format in a group

```
| PRTCAL CAL(*GRP)  
| GRPCALS(GRPCHANGE) DATE(042792)  
| ENDDATE(051292) CALTYP(*GROUP)  
| PRTFMT(*COLUMNS)
```

| This command prints the first date specified in the command for calendars in the group with each column representing one calendar. The other days follow. Seven days of the week should be included without skipping any days.

Example 9: Printing a group calendar in monthly format individually

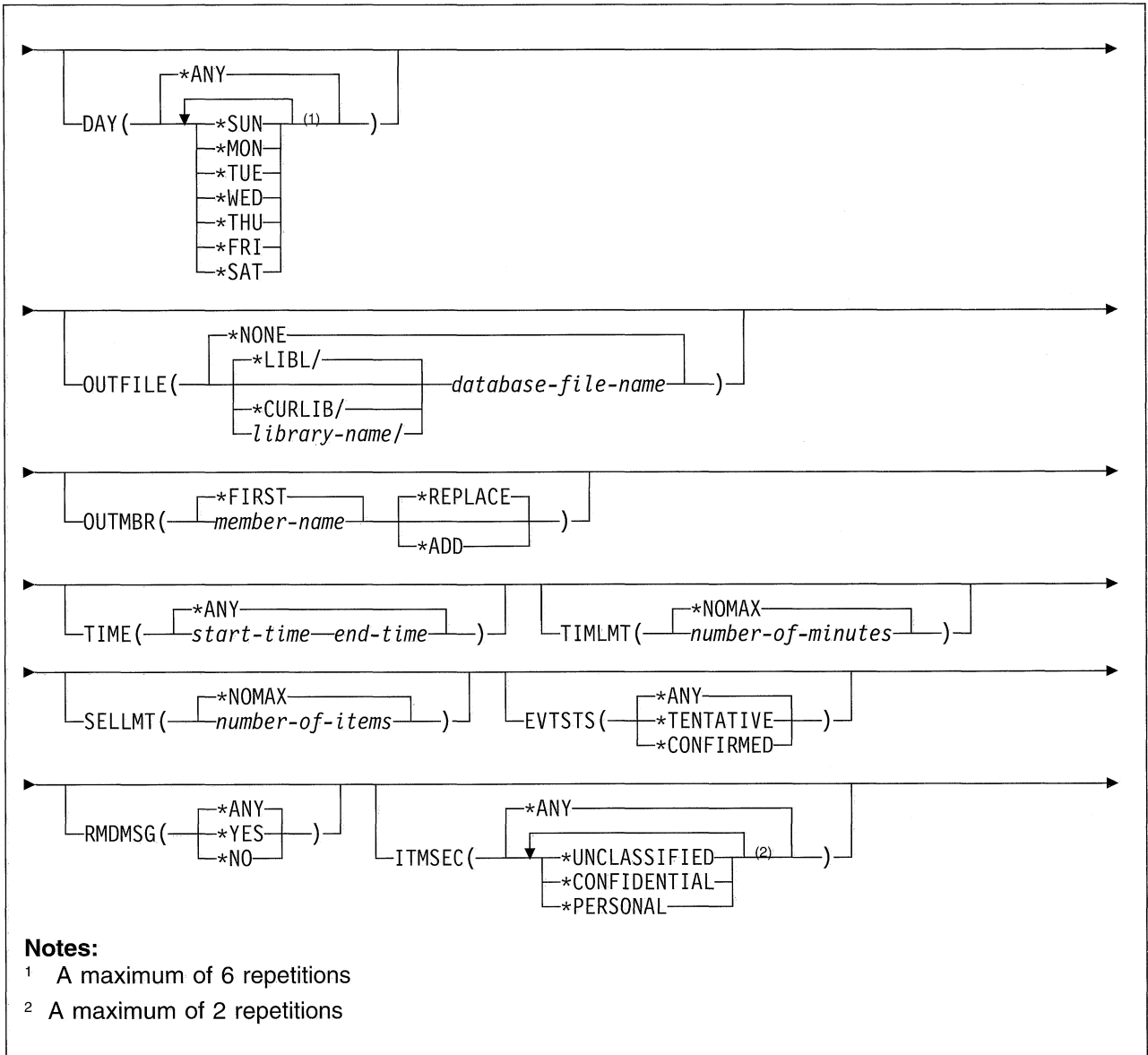
```
| PRTCAL CAL(*GRP)  
|   GRPCALS(GRPCHANGE)  
|   DATE(*CURRENT)  
|   CALTYP(*INDIVIDUAL)  
|   PRPFMT(*MONTHLY)
```

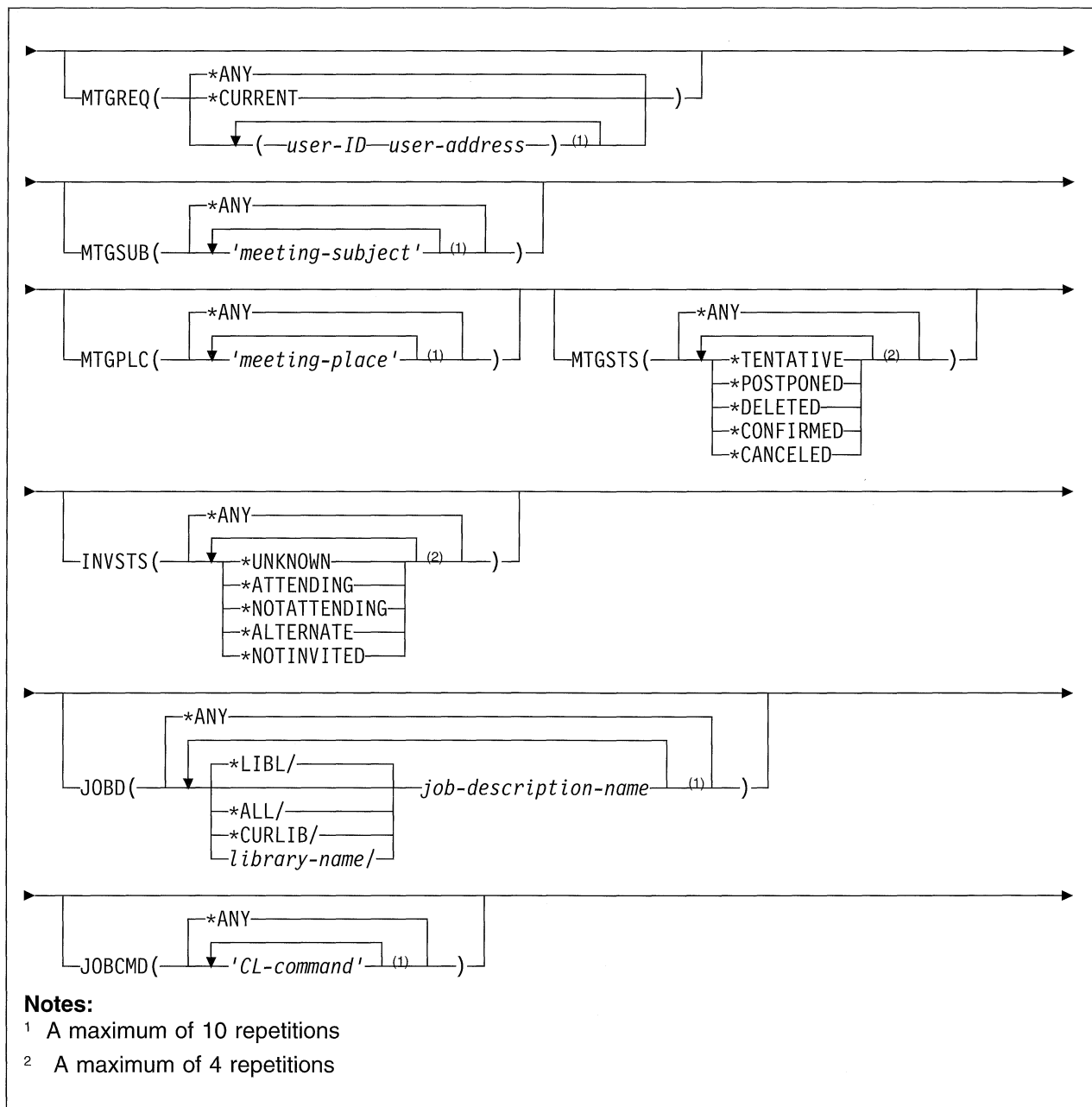
| This command prints a monthly calendar for each
| calendar in the group.

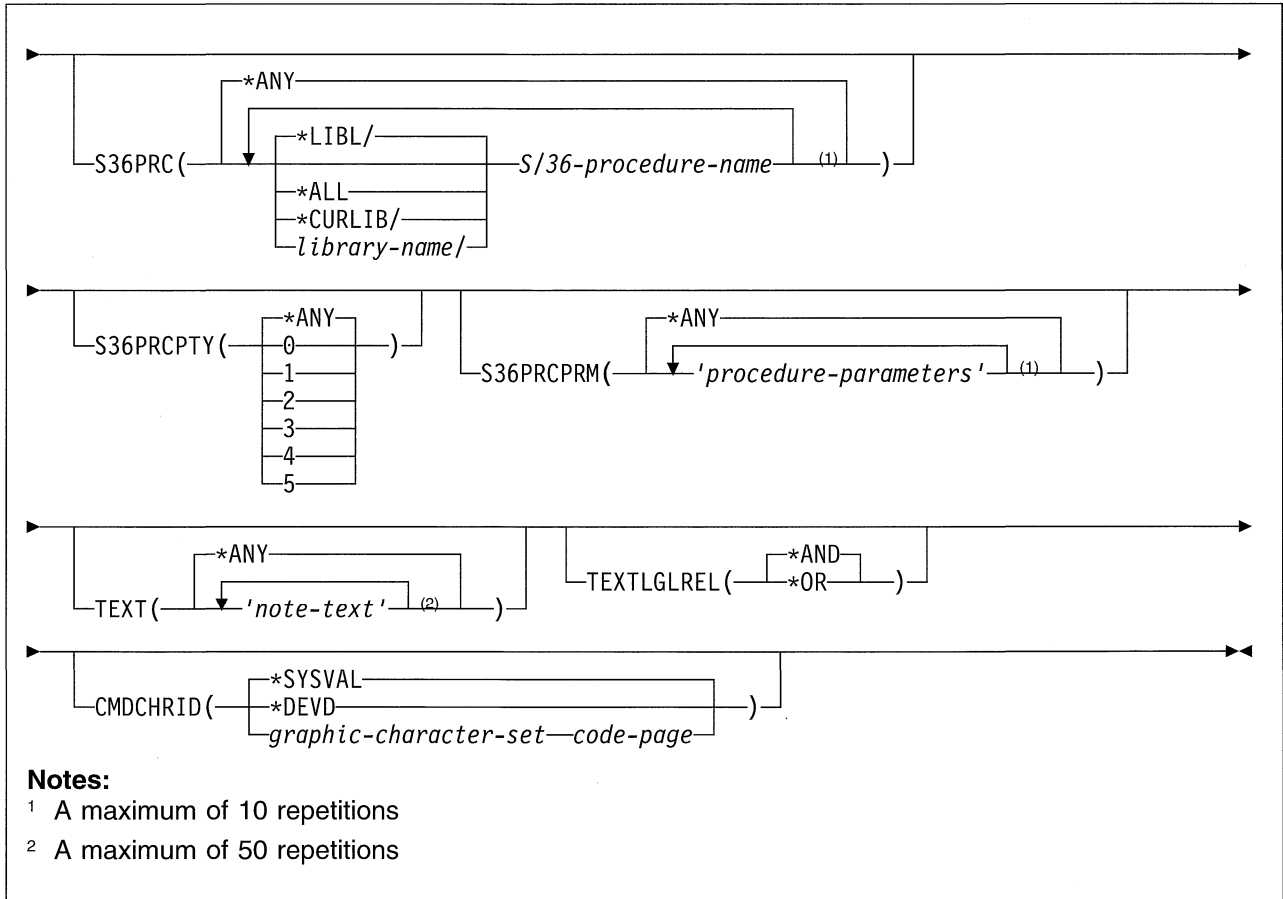
| **Example 10: Printing a group calendar in
| monthly format group**

```
| PRTCAL CAL(*GRP)  
|   GRPCALS(GRPCHANGE)  
|   DATE(*CURRENT) CALTYP(*GROUP)  
|   PRPFMT(*MONTHLY)
```

| This command prints a monthly calendar of the
| group that is similar to the composite view.







Purpose

The Query Calendar Item (QRYCALITM) command allows the user to search one or more calendars for items that match the values specified on this command's parameters. The items returned by the search are directed to a database file. If a database file is not specified, a message that contains a count of the number of items that matched the search values is displayed.

The following chart shows the parameters that apply to each type of calendar item. The search returns items that match the values specified for the parameters listed under each type of item.

Note: If the user has only *TIMES authority to an item, only the CAL, ITMTYP, DAY, DATE, and TIME parameters can be specified on a search. If other parameters are specified, the item does not match during the search.

| EVENT | MEETING | REMINDER | JOB | S/36 PROCEDURE |
|--------|---------|----------|---------|----------------|
| ITMTYP | ITMTYP | ITMTYP | ITMTYP | ITMTYP |
| EVTSTS | RMDMSG | RMDMSG | ITMSEC | ITMSEC |
| RMDMSG | ITMSEC | ITMSEC | JOBID | S36PRC |
| ITMSEC | MTGSUB | DAY | JOB CMD | S36PRCPTY |
| DAY | MTGPLC | DATE | DAY | S36PRCPRM |
| DATE | MTGSTS | TEXT | DATE | DAY |
| TIME | INVSTS | | TIME | DATE |
| TEXT | MTGREQ | | TEXT | TIME |
| | DAY | | | TEXT |
| | DATE | | | |
| | TIME | | | |
| | TEXT | | | |

Parameters

CAL

Specifies the name and owner of the calendar or calendars to search. A calendar name and owner name or *GRP can be specified on this parameter.

Element 1: Calendar Name

***SCHED:** Scheduling calendars are searched.

***ALL:** All calendars are searched.

calendar-name: Specify the name of the calendar to search.

generic-name:* Specify the generic name of the calendar. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all calendars with names that begin with the generic name, and for which the user has authority, are searched. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete calendar name.

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify 'MIKE**' to search all calendars that have names beginning with 'MIKE*'.
|
|

Element 2: Calendar Owner

***CURRENT:** The calendars searched are owned by the user issuing the command.

***ALL:** The calendars searched are owned by local users.

list-identification: Specify a distribution list. Calendars that have a name specified on the first element of this parameter and are owned by users on the distribution list are searched.

user-ID: Specify the user ID of the owner of the calendar(s) searched. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar(s) searched. This value is valid only if user ID is also specified.

Other Single Values

***GRP:** The calendars in the calendar group specified on the GRPCALS parameter are searched.

Note: If *GRP is specified, the GRPCALS parameter must also be specified.

GRPCALS

Specifies the local calendar group to search. This parameter is valid only if CAL(*GRP) is also specified. A value for Element 1 must be specified if GRPCALS is specified.

Element 1: Group Name

group-name: Specify the local calendar group to search.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group to search.

Element 3: User Address

address: Specify the user address of the owner of the calendar group to search.

ITMTYP

Specifies the type of item searched for. *ANY or any combination of up to four item types can be specified.

***ANY:** The search returns all items that match the other search values specified.

***EVENT:** The search returns events.

***REMINDER:** The search returns reminders.

***MEETING:** The search returns meeting information.
|
|

***S36PRC:** The search returns System/36 procedures.

***JOB:** The search returns jobs.

DATE

Specifies the dates on which the items must occur to match the search. This parameter can be used with the DAY parameter to limit the number of items that are returned. For example, if DATE(010190 020190) and DAY(*MON) are specified, the search returns items that occur on Mondays on or between January 1, 1990, and February 1, 1990.

***ANY:** The search returns items regardless of the date on which they occur.

Element 1: Start Date

***CURRENT:** The search returns items that occur on or after the current date and on or before the end date specified on the second element of this parameter. If no end date is specified, the search returns all items that occur on the current date.

***AVAIL:** The search returns items that occur on or before the end date specified on the second element of this parameter. If no end date is specified, the search returns all items that occur on 01/01/1940.

start-date: Specify the start date in the job date format. A maximum of 10 dates can be specified. If no end date is specified, the search returns all items that occur on the start date.

Element 2: End Date

***CURRENT**: The search returns all items that occur on or after the start date specified on the first element of this parameter and on or before the current date.

***END**: The search returns all items that occur on or after the start date.

end-date: Specify the end date in the job date format. A maximum of 10 dates can be specified.

DAY

Specifies the days of the week to search. Items that occur on the days of the week specified are returned. Any combination of days of the week can be specified.

***ANY**: The search returns items regardless of the day of the week on which they occur.

***SUN**: The search returns items that occur on Sundays.

***MON**: The search returns items that occur on Mondays.

***TUE**: The search returns items that occur on Tuesdays.

***WED**: The search returns items that occur on Wednesdays.

***THU**: The search returns items that occur on Thursdays.

***FRI**: The search returns items that occur on Fridays.

***SAT**: The search returns items that occur on Saturdays.

OUTFILE

Specifies the name and library of the database file to which the output of this command is directed. If the file does not exist, the system creates a file in the specified library. If a new file is created, the system uses QAOYQRYCAL in QOFC with a record format name of QRYCAL as a model. If the file already exists, it must have this format.

***NONE**: The output is not directed to a database file. A message is sent that contains a

count of the number of items found that match the search values.

The possible library values are:

***LIBL**: The library list is used to locate the file. If the file is not found, one is created in the current library. If no current library exists, the file is created in the QGPL library.

***CURLIB**: The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the file is located.

database-file-name: Specify the name of the output file.

OUTMBR

Specifies the name of the database file member that receives the output of this command.

Element 1: Member to Receive Output

***FIRST**: The first member in the file receives the output. If it does not exist, the system creates a member with the same name as the file specified on the OUTFILE parameter.

member-name: Specify the name of the file member that receives the output. If it does not exist, the system creates it.

Element 2: Operation to Perform on Member

***REPLACE**: The output data replaces existing records in the specified member.

***ADD**: The output data is added after existing records in the specified member.

TIME

Specifies the time range into which the item must fall to match the search. If the item meets or overlaps the time range specified, the item is returned. For example, if TIME(8:00 10:00) is specified, a meeting that is scheduled to last from 7:00 a.m. to 8:30 a.m. is returned, and a meeting that is scheduled to last from 10:00 a.m. to 10:30 a.m. is returned.

***ANY**: The search returns items regardless of the time at which they occur.

Element 1: Start Time

start-time: Specify the start time.

Element 2: End Time

end-time: Specify the end time.

TIMLMT

Specifies the number of minutes the search can take. If the search reaches the limit specified, the search is canceled. The items found within the time limit that match the search values specified are returned.

***NOMAX:** There is no limit on the number of minutes the search can take.

number-of-minutes: Specify the number of minutes the search can take.

SELLMT

Specifies the number of items matching the search values that are returned. If the number of items reaches the limit specified the search is canceled. The items found within the limit that match the search values are returned.

***NOMAX:** There is no limit on the number of items that are returned.

number-of-items: Specify the number of items that can be returned before the search is canceled.

EVTSTS

Specifies the status of the item searched for.

Note: This parameter is valid only if *EVENT or *ANY is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of their status.

***TENTATIVE:** The search returns only tentative events. Tentative events have dates and times that may change.

***CONFIRMED:** The search returns only confirmed events. Tentative events have dates and times that will not change.

RMDMSG

Specifies whether the item searched for has a reminder sent to the owner of the calendar before the item is scheduled to occur.

Note: This parameter is valid only if *ANY, *EVENT, *MEETING, or *REMINDER is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of whether a reminder is sent.

***YES:** The search returns items for which a reminder is sent.

***NO:** The search returns items for which no reminder is sent.

ITMSEC

Specifies the security level of the item searched for. Any combination of security levels can be specified.

***ANY:** The search returns items regardless of security level.

***UNCLASSIFIED:** The search returns items that are unclassified.

***CONFIDENTIAL:** The search returns items that are confidential.

***PERSONAL:** The search returns items that are personal.

MTGREQ

Specifies the user ID and address of the person who requested the item searched for.

Note: This parameter is valid only if *ANY or *MEETING is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of the person who requested them.

user-ID: Specify the user ID. This value is valid only if user address is also specified. Up to 10 user IDs can be specified.

user-address: Specify the user address. This value is valid only if user ID is also specified. Up to 10 user addresses can be specified.

Other Single Values

***CURRENT:** The search returns meetings requested by the current user.

MTGSUB

Specifies the subject of the item searched for.

Note: This parameter is valid only if *ANY or *MEETING is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of their meeting subject.

'meeting-subject': Specify the meeting subject. Up to 10 meeting subjects can be specified.

MTGPLC

Specifies the meeting place of the item searched for.

Note: This parameter is valid only if *ANY or *MEETING is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of their meeting place.

'meeting-place': Specify the meeting place. Up to 10 meeting places can be specified.

MTGSTS

Specifies the status of the item searched for. Any combination of meeting statuses can be specified.

Note: This parameter is valid only if *ANY or *MEETING is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of their status.

***TENTATIVE:** The search returns items with dates and times that could change.

***POSTPONED:** The search returns items that will be changed to a later date.

***DELETED:** The search returns items that have been deleted.

***CONFIRMED:** The search returns items with dates and times that will not change.

***CANCELED:** The search returns items that have been canceled.

INVSTS

Specifies the invitee status of the items searched for. Any combination of statuses can be specified.

Note: This parameter is valid only if *ANY or *MEETING is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of their invitee status.

***UNKNOWN:** The search returns items with invitee status unknown.

***ATTENDING:** The search returns items that the invitee is attending.

***NOTATTENDING:** The search returns items that the invitee is not attending.

***ALTERNATE:** The search returns items that the invitee is not attending, but is sending an alternate attendee.

***NOTINVITED** The search returns meetings that appear on the calendar specified on the CAL parameter, but for which the calendar owner is not an invitee.

JOB

Specifies the name and library of the job description of the item searched for.

Note: This parameter is valid only if *ANY or *JOB is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of their job description.

The possible library values are:

***LIBL:** The library list is used to locate the job description.

***ALL:** All libraries in the system, including QSYS, are searched.

***CURLIB:** The current library for the job is used to locate the job description. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the job description is located.

job-description-name: Specifies the name of the job description. Up to 10 names can be specified.

JOBCMD

Specifies the name of the CL command to be run in the scheduled job searched for. If the command string specified matches any part of a scheduled job's CL command, the search returns that scheduled job. For example, if USRPRF is specified on this parameter, the search returns any scheduled jobs whose CL command contains the character string USRPRF, such as CHGUSRPRF.

Note: This parameter is valid only if *ANY or *JOB is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of the name of the CL command.

'CL-command': Specify the CL command name. Up to 10 command names can be specified.

S36PRC

Specifies the name and library of the System/36 procedure searched for.

Note: This parameter is valid only if *ANY or *S36PRC is specified on the ITMTYP parameter.

***ANY:** The system returns items regardless of whether they are a System/36 procedure.

The possible library values are:

***LIBL:** The library list is used to locate the procedure.

***ALL:** All libraries in the system, including QSYS, are searched.

***CURLIB:** The current library for the job is used to locate the procedure. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the procedure is located.

S/36-procedure-name: Specify the name of the System/36 procedure. A maximum of 10 procedures can be specified.

S36PRCPTY

Specifies the priority of the System/36 procedure searched for. The highest priority is 5; the lowest priority is 0. A maximum of 10 priority levels can be specified.

Note: This parameter is valid only if *ANY or *S36PRC is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of their priority level.

- 0: The priority is 0.
- 1: The priority is 1.
- 2: The priority is 2.
- 3: The priority is 3.
- 4: The priority is 4.
- 5: The priority is 5.

S36PRCPRM

Specifies the parameters to be passed to the System/36 procedure searched for.

Note: This parameter is valid only if *ANY or *S36PRC is specified on the ITMTYP parameter.

***ANY:** The search returns items regardless of the parameters to be passed.

'procedure-parameters': Specify the procedure parameters to be passed. Up to 10 parameters can be specified.

TEXT

Specifies text that briefly describes the object. More information on this parameter is in "Appendix A. Expanded Parameter Descriptions".

***ANY:** The search returns items regardless of text.

'note text': Specify the item text. Up to 50 characters can be specified.

TEXTLGLREL

Specifies how an item matches the search values.

***AND:** To match the search values, an item must contain all of the text specified on the TEXT, MTGSUB, and MTGPLC parameters.

***OR:** To match the search values, an item must contain any of the text specified on the TEXT, MTGSUB, or MTGPLC parameters.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device is used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page value.
Valid values range from 1 through 9999.

Examples

Example 1: Searching a Scheduling Calendar

```
QRYCALITM CAL(*SCHED) DATE(030190)
      TIME(('8:00' '10:00'))
```

This command searches the scheduling calendar of the user issuing the command for all items between 8:00 a.m. and 10:00 a.m. on March 1, 1990.

Example 2: Searching All Calendars

```
QRYCALITM CAL(*ALL CONFMS RCHAS1)
      DATE(030190) TIME(('8:00' '10:00'))
```

This command searches all the calendars owned by CONFMS RCHAS1. It will return all items scheduled between 8:00 a.m. and 10:00 a.m. on March 1, 1990.

Example 3: Searching for a Scheduled Job

```
QRYCALITM CAL(*ALL *ALL) ITMTYP(*JOB *S36PRC)
      JOBCMD(SAVLIB) S36PRC(SAVLIBR)
      OUTFILE(MYLIB/JOBQRY)
```

This command searches all the calendars on the system for a scheduled job involving a System/36 procedure. The command performs the Save Library (SAVLIB) command on the scheduled SAVLIBR System/36 procedure. The results are stored in the output file JOBQRY in library MYLIB.

with names that begin with the generic name, and for which the user has authority, are removed. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete calendar name.

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify 'MIKE**' to remove the item from all calendars that have names beginning with 'MIKE*'.

Element 2: Calendar Owner

***CURRENT:** The item is removed from the specified calendars owned by the user of the command.

***ALL:** The item is removed from all specified calendars that are owned by local users.

Note: The user must have *SECADM or *ALLOBJ special authority to specify *ALL for both the calendar name and the owner.

list-identification: Specify a distribution list. The item is removed from all specified calendars that are owned by users on the distribution list.

user-ID: Specify the user ID of the owner of the calendars from which the item is removed. This value is valid only if user address is also specified.

user-address: Specify the address of the owner of the calendars from which the item is removed. This value is valid only if user ID is specified.

GRPCALS

Specifies the local calendar group from which the item is removed. This parameter is valid only if CAL(*GRP) is also specified. A value for Element 1 must be specified if GRPCALS is specified.

Element 1: Group Name

group-name: Specify the local calendar group from which the item is removed.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group.

Element 3: User Address

user-address: Specify the user address of the owner of the calendar group.

DATE

Specifies the dates between which items are removed from the specified calendars. All items that occur on or between the specified dates are removed.

Note: The user must specify one of these parameters: DATE, ITIMAGE, or ITMID. When one is specified, the other two are not allowed. If a value is not specified for one of these parameters, the DATE parameter default values are used.

Element 1: Start Date

***AVAIL:** All items that occur on or before the ending date are removed from the specified calendars.

start-date: Specify the starting date. The date must be specified in the job date format.

Element 2: End Date

***CURRENT:** All items that occur on or after the starting date and on or before the current date are removed.

end-date: Specify the ending date. The date must be specified in the job date format.

ITIMAGE

Specifies the age, in days, of the items removed. Items as old as or older than the number of days specified are removed. For example, if 0 is specified, all items that occur on the current date or occurred before the current date are removed.

Note: The user must specify one of these parameters: DATE, ITIMAGE, or ITMID. When one is specified, the other two are not allowed. If a value is not specified for one of these parameters, the DATE parameter default values are used.

ITMID

Specifies the item identifiers of the items to be removed. The item identifier can be a maximum of 20 characters. This parameter is

RMVCALITM

valid only when a single calendar is specified on the CAL parameter. A maximum of 50 item identifiers can be specified.

Note: The user must specify one of these parameters: DATE, ITIMAGE, or ITMID. When one is specified, the other two are not allowed. If a value is not specified for one of these parameters, the DATE parameter default values are used.

RGZPFM

Specifies whether the database file members containing the calendar data are reorganized after the items are removed. Reorganizing the database file recovers the space used by the deleted records and improves performance. The database files reorganized are: QAOCIT5, QAOTXT5, QAOCIN5, QAOCCCL5, QAOCAU5, QAOCGR5, and QAOCGM5 in library QUSRSYS.

Note: If RGZPFM(*YES) is specified, no other users can be using calendar operations. If other users are using calendar operations, the remove operation will work but the reorganize operation will not work.

***NO:** The database file member is not reorganized.

***YES:** The database file member is reorganized.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEVVD:** The character set and code page defined for the device is used. This value cannot be specified if the command is run in batch.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples

Example 1: Removing One Item

```
RMVCALITM CAL(*SCHED) ITMID(&ITMID)
```

This command removes the item that is contained in the CL variable &ITMID from the scheduling calendar of the user issuing the command.

Example 2: Removing All Items by Date

```
RMVCALITM CAL(*ALL *CURRENT) DATE(*AVAIL 123189)
```

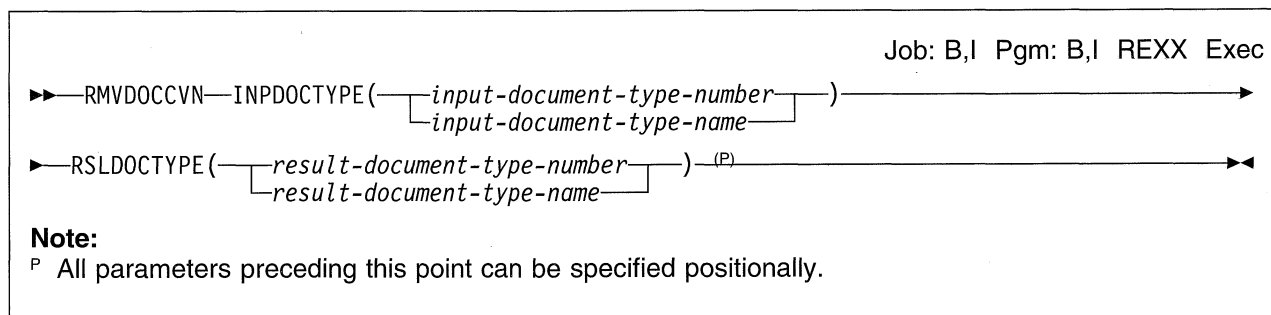
This command removes all items from all calendars owned by the user issuing the command that have a date before January 1, 1990.

Example 3: Removing All Items and Recovering Space

```
RMVCALITM CAL(*ALL *ALL) ITIMAGE(30) RGZPFM(*YES)
```

This command removes all items from all calendars on the system that are 30 days old or older. The space used by the deleted records is recovered.

RMVDOCCVN (Remove Document Conversion) Command



Purpose

The Remove Document Conversion (RMVDOCCVN) command removes a document conversion definition.

Restriction: You must have security officer (*SECOFR) authority or be enrolled in OfficeVision/400 as an administrator to use this command.

Parameters

INPDOCTYPE

Specifies the name or number of the document type used as input for the document conversion definition being removed.

input-document-type-number: Specify the numeric value associated with the Document Interchange Architecture (DIA) document type requested. Valid values range from 1 through 65535.

input-document-type-name: Specify the 10-character name associated with the defined document type.

RSLDOCTYPE

Specifies the name or number of the document type used as the result type for the document conversion definition being removed.

result-document-type-number: Specify the numeric value associated with the DIA document type requested. Valid values range from 1 through 65535.

result-document-type-name: Specify the 10-character name associated with the defined document type.

Example

```
RMVDOCCVN INPDOCTYPE(RFTDCA) RSLDOCTYPE(32776)
```

This command removes the specified document conversion.

RMVOFCAPPE (Remove Office Application Entry) Command

Job: B,I Pgm: B,I REXX Exec

```

  ▶ RMVOFCAPPE SET (—*ALL—set-ID—) —USRPRF (—
    [ *CURRENT —
    [ *SYSTEM —
    [ user-profile-name —
  ) (P) ▶
  
```

Note:
 P All parameters preceding this point can be specified positionally.

Purpose

The Remove Office Application Entry (RMVOFCAPPE) command allows the user to remove an association between office applications and sets for a specified user.

Note: The user must sign off in order for the removed entry to go into affect.

Parameters

SET

Specifies the set to be removed from association with the user.

***ALL:** All application entries for the identified user will be removed. If no entries exist, the completion message will be 0 for the count of entries deleted.

set-ID: The ID of an existing set associated with an application.

USRPRF

Specifies the user for whom office application entries are removed.

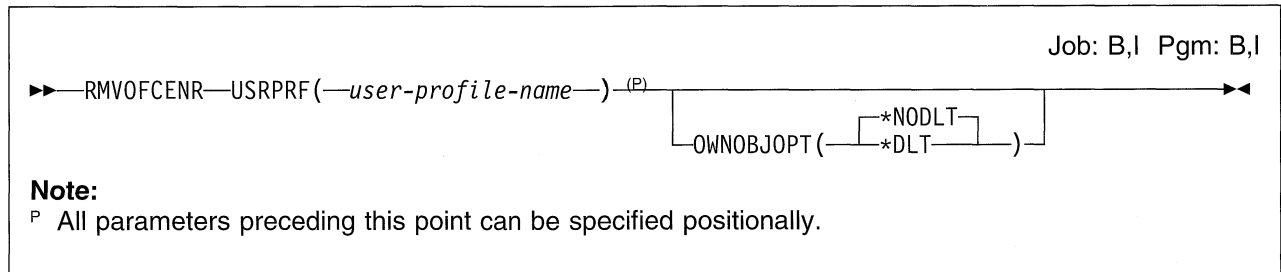
***CURRENT:** The application entries for the current signed on user are removed.

Note: When the user does not have *ALLOBJ or *SECADM authority, *CURRENT is the only valid value for this parameter.

***SYSTEM:** The system default values are used if the user has no requested application for a set.

user-profile-name: Specify the name of the user profile used.

RMVOFCENR (Remove Office Enrollment) Command



Purpose

The Remove Office Enrollment (RMVOFCENR) command removes a user's enrollment information from OfficeVision/400. This command does not affect the user's System Distribution Directory entry or the user's user profile on the AS/400 system. It does not remove calendars if the calendar option is not installed.

Restriction: The user of this command must be designated a security officer and be enrolled in OfficeVision/400 as an administrator.

Parameters

USRPRF

Specifies the name of the user profile of the user being removed from OfficeVision/400.

OWNOBJOPT

Specifies whether the objects owned by the user profile of the user being removed are deleted.

***NODLT:** If the user profile owns objects, the objects are not deleted and the user is not removed from OfficeVision/400.

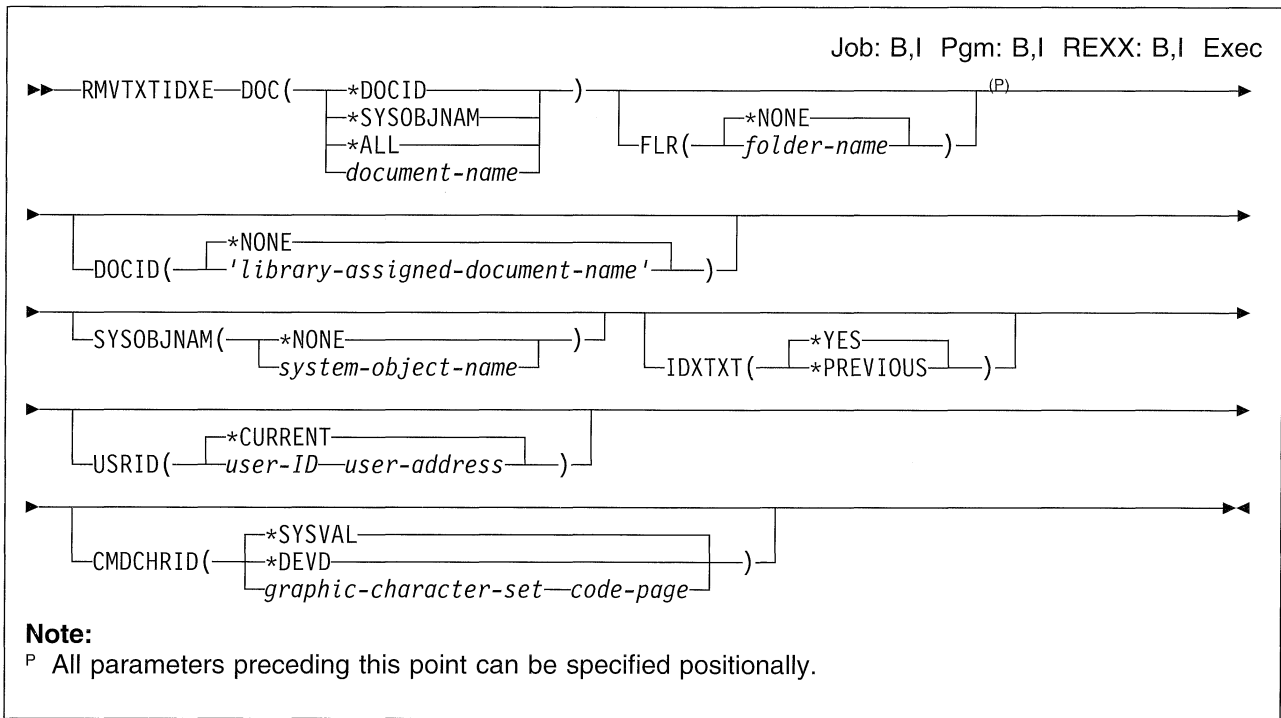
***DLT:** The objects owned by the user profile are deleted. If the deletion of the objects is successful, the user's enrollment information is removed from OfficeVision/400.

Example

```
RMVOFCENR USRPRF(JACKSON) OWNOBJOPT(*DLT)
```

This command removes the enrollment information of user profile JACKSON and all user-owned calendars and directories from OfficeVision/400.

RMVTXTIDX (Remove Text Index Entry) Command



Purpose

The Remove Text Index Entry (RMVTXTIDX) command removes a document's index entry from the text search index. Index entries are removed during the next scheduled index update. Use the Display Index Status (DSPIDXSTS) command to display the update schedule. If a document's index entry is removed from the text search index, the document can no longer be located by using the QRYTXT parameter of the Query Document Library (QRYDOCLIB) command or by specifying document text criteria when searching for documents with OfficeVision/400.

Index entries can be removed by specifying:

- The document name (DOC) and the folder (FLR) in which the document exists
- The library-assigned document name using DOC(*DOCID) and the DOCID parameter
- The system object name using DOC(*SYSOBJNAM) and the SYSOBJNAM parameter

The index entries of all documents in a folder can be removed by specifying DOC(*ALL) and the name of the folder on the FLR parameter.

Restriction: The user, or the user for whom the request is being made, must have at least *USE authority for the document or *ALLOBJ special authority.

Parameters

DOC

Specifies the document for which the index entry is removed.

***DOCID:** The document specified on the DOCID parameter is used.

***SYSOBJNAM:** The document specified on the SYSOBJNAM parameter is used.

***ALL:** Index entries are removed for all documents in the folder specified on the FLR parameter.

document-name: Specify up to 12 characters for the name of the document for which the index entry is to be removed.

FLR

Specifies the name of the folder that contains the documents for which index entries are removed. A folder name must be specified if a user-assigned document name or *ALL is specified on the DOC parameter.

***NONE:** This value is used if *DOCID or *SYSOBJNAM is specified on the DOC parameter.

folder-name: Specify a maximum of 63 characters for the user-assigned name of the folder containing the document. If the document is nested within folders, all of the associated folders must be specified in the form FLR1/FLR2/FLRn, where FLRn is the folder in which the document exists.

DOCID

Specifies the library-assigned name of the document for which the index entry is removed. A library-assigned document name must be specified if *DOCID is specified on the DOC parameter.

A library-assigned document name is a system-generated name based on the date and time that the document was created. Library-assigned document names can be determined by using the Query Document Library (QRYDOCLIB) command or by the message returned from the File Document (FILDOC) command.

Library-assigned document names are 24 characters in length with the following format: YYYYMMDDHHMNSSHSSNSNSNSN where:

- YYYY = year
- MM = month
- DD = day
- HH = hour
- MN = minute
- SS = second
- HS = hundredths of a second
- SNSNSNSN = system name

***NONE:** No library-assigned document name is specified.

'library-assigned-document-name': Specify the library-assigned name of the document.

SYSOBJNAM

Specifies the system object name of the document for which the index entry is removed. A

system object name must be specified if *SYSOBJNAM is specified on the DOC parameter.

***NONE:** No system object name is specified.

system-object-name: Specify the system object name of the document.

USRID

Specifies the user ID and user address associated with the request.

Note: To work on another's behalf, you must have been given permission with the Grant User Permission (GRTUSRPMN) command.

***CURRENT:** You are making the request for yourself.

Element 1: User ID

user-ID: Specify your user ID or the user ID of the user for whom you are making the request.

Element 2: User Address

user-address: Specify your user address or the user address of the user for whom you are making the request.

CMDCHRID

Specifies the graphic character set and code page of the characters entered as parameter values. The graphic character set and code page are related to the type of display device used.

***SYSVAL:** The graphic character set and code page are taken from the QCHRID system value.

***DEVd:** The graphic character set and code page are taken from the display device description. This value is valid only in the interactive environment.

Element 1: Character Set

graphic-character-set: Specify the graphic character set. Valid values range from 1 through 999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 999.

RMVXTIDXE

Examples

Example 1: Removing a Document's Index Entry

```
RMVXTIDXE DOC(ABC) FLR(XYZ)
```

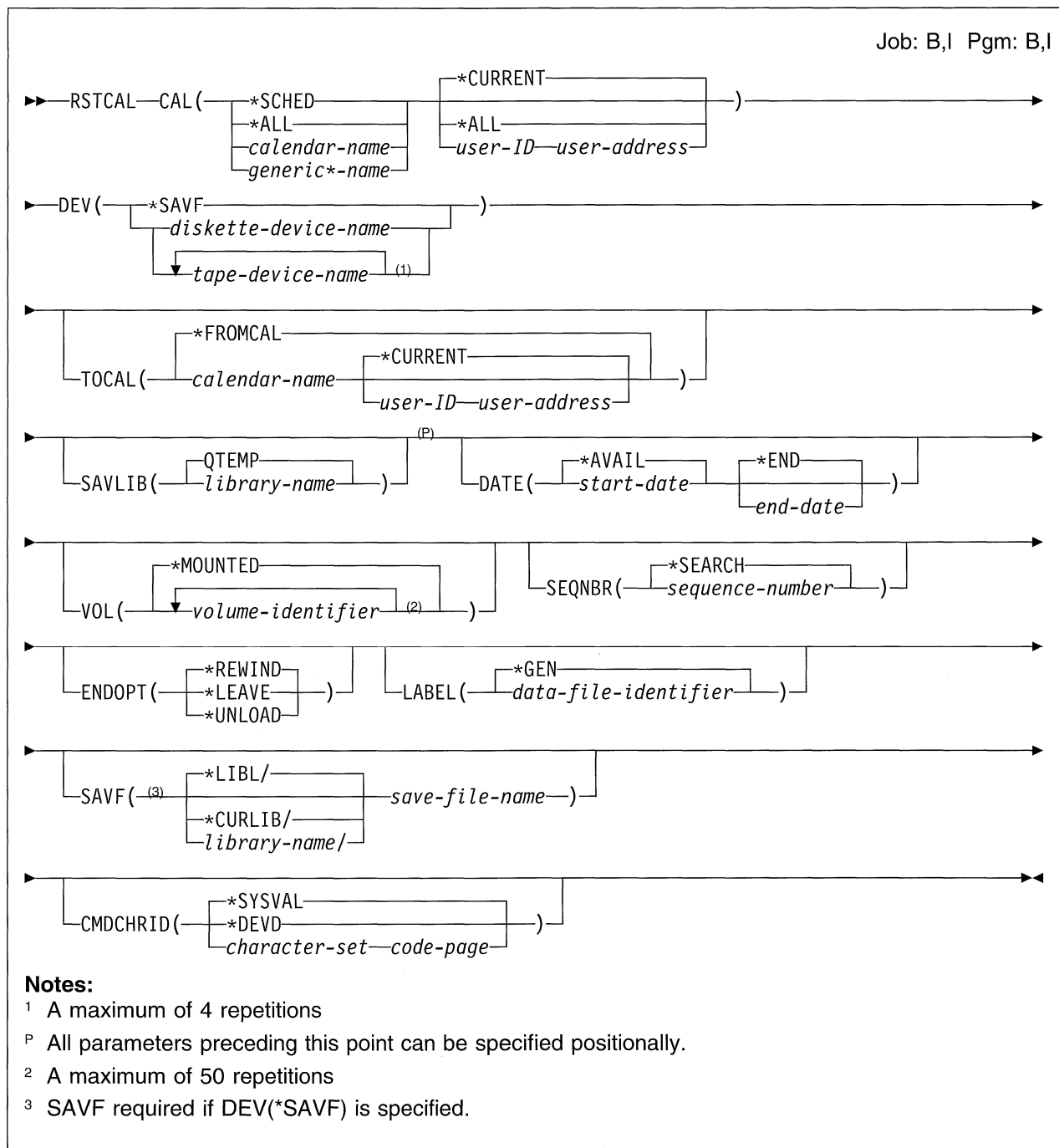
This command removes the index entry of document ABC, which exists in folder XYZ, from the text search index.

Example 2: Removing Index Entries for all Documents in a Folder

```
RMVXTIDXE DOC(*ALL) FLR(XYZ)
```

This command removes the index entries of all documents in folder XYZ from the text search index.

RSTCAL (Restore Calendar) Command



Purpose

The Restore Calendar (RSTCAL) command allows you to restore calendars that were previously saved as part of a system backup of library QUSRSYS, or that were saved using the SAVCAL command. You may restore one or more calen-

dars to the calendar they were saved from, or you may restore a calendar to a different calendar. If the calendar does not exist, it is created during the restore function.

Note: Calendars only can be restored from a system backup of the QUSRSYS library in Version 2 Release 1 Modification 0 or sub-

sequent releases. A system backup of the QUSRSYS library from earlier releases cannot be used.

Parameters

CAL

Specifies the name and owner of the calendar to be restored.

Restriction: The user must have *USE authority for the calendar to restore the items on the calendar, or have *SAVSYS, *SECADM, or *ALLOBJ authority.

Element 1: Calendar Name

***SCHED:** The scheduling calendars owned by the users specified in the second element of this parameter are restored.

***ALL:** All calendars owned by the users specified on the second element of this parameter are restored.

calendar-name: Specify the name of the calendar to restore.

generic-name:* Specify the generic name of the calendar to restore. A generic name is a character string of one or more characters followed by an asterisk (*).

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify 'MIKE**' to restore all calendars that have names beginning with 'MIKE*'. |

Element 2: Calendar Owner

***CURRENT:** The named calendars owned by the user issuing the command are restored. |

***ALL:** All named calendars owned by local users are restored. |

user-ID: Specify the user ID of the owner of the calendar to be restored. This value is valid only if user address is also specified. |

user-address: Specify the user address of the owner of the calendar to be restored. This value is valid only if user ID is also specified. |

TOTAL

Specifies the name and owner of the calendar to which the calendar specified on the CAL parameter is restored.

Note: You must have *ADDITEM authority for the calendar or have *SAVSYS, *SECADM, or ALLOBJ authority to restore the items to the calendar.

This parameter is valid only if a single, local calendar is specified on the CAL parameter.

Element 1: Calendar Name

calendar-name: Specify the name of the calendar to which the calendar is restored.

Element 2: Calendar Owner

***CURRENT:** The user issuing the command is the owner of the calendar. |

user-ID: Specify the user ID of the owner of the calendar to be restored. This value is valid only if user address is also specified. |

user-address: Specify the user address of the owner of the calendar to be restored. This value is valid only if user ID is also specified. |

Other Single Value

***FROMCAL:** The calendar is restored to the same calendar from which it was saved. If the specified calendar does not exist, it is created.

SAVLIB

Specifies the name of the library from which the calendar database files are saved.

QTEMP: If calendars are saved with the SAVCAL command, the library QTEMP is used.

QUSRSYS: If the calendars are saved by specifying SAVLIB LIB(QUSRSYS) or SAVLIB LIB(*NONSYS), the library QUSRSYS is used. |
If the calendars are saved by using the SAVCAL command and specifying CAL(*ALL *ALL) and DATE(*AVAIL *END), the library QUSRSYS is used. |

library-name: Specify the name of the library. |

DEV

Specifies the name of the device from which the calendar is restored. This is a required parameter.

***SAVF:** The calendar is restored from a save file.

diskette-device-name: Specify the name of the diskette device from which the calendar is restored. Up to 4 diskette device names can be specified.

tape-device-name: Specify the name of the tape device from which the calendar is restored. Up to 4 tape device names can be specified.

DATE

Specifies a range of dates within which all items on the calendar, including items that occur on the start date and end date, are restored.

Note: If ***AVAIL** is specified for the start date and ***END** for the end date, all saved items on the calendars are restored.

Element 1: Start Date

***AVAIL:** All calendars before the end date specified are restored.

start-date: Specify a start date.

Element 2: End Date

***END:** All calendars after the start date specified are restored.

end-date: Specify an end date.

VOL

Specifies the volume identifiers of the tape volumes or diskette volumes from which the objects are being restored. The volumes must be in the same order as they were when the library was saved. If the library was known to the system before the restore operation, the system can verify whether the volumes put on tape contain the current version of the library. The volume that contains the beginning of the file to be restored should be placed in the tape or diskette unit.

***MOUNTED:** The volume identifier of the tape or diskette placed on the device is used.

volume-identifier: Specify the volume identifier. A maximum of 50 volume identifiers can be specified.

SEQNBR

Specifies, for a tape device, the sequence number of the file from which the calendars are restored.

***SEARCH:** The tape is searched from its current position for the label specified on the LABEL parameter.

sequence-number: Specify the sequence number on the tape to which the calendars are restored.

ENDOPT

Specifies, for a tape device, whether the tape is rewound, left, or rewound and unloaded after the calendar is restored.

***REWIND:** The tape is rewound.

***LEAVE:** The tape is left as it is.

***UNLOAD:** The tape is rewound and unloaded.

LABEL

Specifies the label that identifies the data file on the tape or diskette used for the restore operation.

If the SAVCAL command is used for the save, and if ***GEN** is used, the label is generated using the calendar name and owner. The Display Tape (DSPTAP) command can be used to display the label.

***GEN:** The system generates the label name.

Note: Use this value only if you are restoring a calendar which was saved using the SAVCAL command.

data-file-identifier: Specify the label name.

SAVF

Specifies the qualified name of the save file from which the calendar is restored.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the file is located.

save-file-name: Specify the name of the save file from which the calendar is restored.

RSTCAL

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEV D:** The character set and code page defined for the device is used. This value cannot be used when running the command in batch.

Element 1: Character Set

character-set: Specify the character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

Examples

Example 1: Restoring a Scheduling Calendar

```
RSTCAL CAL(*SCHED) DEV(TAP01)
```

This command restores the scheduling calendar of the user issuing the command. The calendar is restored from the tape device TAP01. All calendar items are restored.

Example 2: Restoring All Items by Date

```
RSTCAL CAL(*ALL *ALL) DATE(010190 *END)
DEV(TAP01)
```

This command restores all the items on all the calendars on or after January 1, 1990. The calendars are restored from the device named TAP01. The user issuing this command must be a security administrator.

Example 3: Restoring a Calendar to Another Calendar

```
RSTCAL CAL(*SCHED PJC RCHAS1)
TOCAL(*SCHED *CURRENT) DEV(*SAVF)
SAVF(RMSLIB/CALSAVF)
```

This command restores PJC's scheduling calendar to the scheduling calendar of the user issuing the command. The calendar is restored from a save file named CALSAVF in library RMSLIB. All calendar items are restored.

RTVOFCA

- | ***INST:** The contextual search option of OfficeVision/400 is installed.
- | ***NOTINST:** The contextual search option of OfficeVision/400 is not installed.

EDTOPT

- | Specifies the name of the CL variable that receives the OfficeVision/400 value that determines whether the OfficeVision/400 editor option is installed. The variable must be at least 8 characters in length. The possible values for the variable are:

- | ***INST:** The editor option of OfficeVision/400 is installed.
- | ***NOTINST:** The editor option of OfficeVision/400 is not installed.

Example

```
RTVOFCA AUTOENR(&AUTOVAL)
```

This command retrieves the current OfficeVision/400 value that determines whether to automatically enroll users when OfficeVision/400 services are requested. A *YES or *NO value is retrieved to the CL variable when the command is run.

```
| RTVOFCA EDTOPT(&EDTOPT)
```

| This command retrieves the current OfficeVision/400 value which determines whether the editor option of the OfficeVision/400 product is installed on the system. The CL variable will contain either *INST or *NOTINST after the command runs.

USRPGMLIB

Specifies the name of the CL variable that receives the name of the library that locates the user program when the user selects Option 50 on the OfficeVision/400 menu. The existence of the specified user program is not verified. The user program cannot require any parameters. The variable must be 10 characters in length.

USRPGMTEXT

Specifies the name of the CL variable that receives the name of the text when the user selects Option 50 on the OfficeVision/400 menu. The existence of the specified user program is not verified. The user program cannot require any parameters. The variable must be 40 characters in length.

ALWDOCCMD

Specifies the name of the CL variable that receives the value that determines whether the user can use CL commands when working with text documents. The variable must be at least 10 characters in length.

WRDPROC

Specifies the name of the CL variable that receives the value that determines the type of word processing function used. The variable must be at least 10 characters in length.

INLPRSDIR

Specifies the name of the CL variable that receives the name of the first personal directory the user will use when running personal directory functions. If a value of *NONE is returned, the user must select a personal directory each time the personal directory function is used. The variable must be at least 10 characters in length.

DFTFLR

Specifies the name of the CL variable that receives the name of the folder that is used whenever a folder name is needed but not specified. The variable must be at least 63 characters in length.

INLCAL

Specifies the name of the CL variable that receives the name of the first calendar the user sees when using calendar functions. The name of a calendar, calendar group, distribution list calendar, or user ID and address

is returned. When the user ID and address is returned, the scheduling calendar for the user ID becomes the current user's initial calendar. The variable must be at least 28 characters in length.

SCHEDCAL

Specifies the name of the CL variable that receives the name of the user's scheduling calendar. The variable must be at least 10 characters in length.

CALFMT

Specifies the name of the CL variable that receives the type of view shown when only one calendar is used, and the type of view shown when more than one calendar is used. The variable must be at least 20 characters in length.

CALTIME

Specifies the name of the CL variable that receives the time of day at which the calendar starts, the format in which the time is shown, and the number of minutes in each time slot on the calendar. The variable must be at least 10 characters in length.

Note: If the variable is not 14 characters long, the number of minutes in each time slot on the monthly display is not returned.

CALSTRDAY

Specifies the name of the CL variable that receives the day of the week shown in the left column of the weekly calendar, and the day of the week shown in the left column of the six-month calendar. The variable must be at least 9 characters in length.

Note: If the variable is not 14 characters long, the day of the week in the left column of the monthly display is not returned.

CALNBRCOL

Specifies the name of the CL variable that receives the number of columns that are shown on weekly, group, monthly or composite views of the calendar, and the number of days that are shown on the daily calendar. Depending on the value returned on the CALFMT parameter, each column represents one of the following:

- A consecutive day for a weekly calendar

- A different user for the same day for a group calendar
- A consecutive day containing calendar items for up to 10 users for a composite calendar.

| The variable must be at least five characters
| in length. If the variable is not seven charac-
| ters long, the number of columns used on the
| monthly display is not returned.

ALWCALJOB

Specifies the name of the CL variable that receives the type of the jobs allowed on the

calendar. The variable must be 10 characters in length.

| **DSPWKNBR**

| Specifies the name of the CL variable that dis-
| plays the week number on the calendar. The
| variable must be 10 characters in length.

Example

```
RTVOFCENR USRPRF(*CURRENT) WRDPROC(&WRDPROC)
```

This command retrieves the current user and the type of word processing the user uses.

Purpose

The Save Calendar (SAVCAL) command is used to save calendars and calendar items. Only local calendars can be saved.

Parameters

CAL

Specifies the name and owner of the calendar to be saved. A calendar name and owner name, or *GRP, can be specified on this parameter.

Restriction: The user must have *USE authority for all security levels on the calendar or have *SECADM, *ALLOBJ, or *SAVSYS special authority to save the items on the calendar.

Element 1: Calendar Name

***SCHED:** The scheduling calendars owned by the users specified in the second element of this parameter are saved.

***ALL:** All calendars owned by the users specified on the second element of this parameter are saved.

calendar-name: Specify the name of the calendar to be saved.

generic-name:* Specify the generic name of the calendar. A generic name is a character string of one or more characters followed by an asterisk (*); for example, ABC*. If a generic name is specified, then all calendars with names that begin with the generic name, and for which the user has authority, are saved. If an asterisk is not included with the generic (prefix) name, the system assumes it to be the complete calendar name.

Note: Any calendar name that ends in an asterisk is assumed to be a generic name. If the last character in a calendar name is an asterisk, add another asterisk to indicate that it is a generic name. For example, specify 'MIKE**' to save all calendars that have names beginning with 'MIKE*'.

Element 2: Calendar Owner

***CURRENT:** The named calendars owned by the user issuing the command are saved.

***ALL:** All named calendars owned by local users are saved.

list-identification: Specify a distribution list. All named calendars owned by users on the distribution list are saved.

user-ID: Specify the user ID of the owner of the calendar to be saved. This value is valid only if user address is also specified.

user-address: Specify the user address of the owner of the calendar to be saved. This value is valid only if user ID is also specified.

***GRP:** All of the calendars in the calendar group specified on the GRPCALS parameter are saved.

Note: If *GRP is specified, the GRPCALS parameter must also be specified.

GRPCALS

Specifies the name of the calendar group containing the calendars to be saved. This parameter is valid only if CAL(*GRP) is also specified. A value for Element 1 is required if GRPCALS is specified.

Note: The calendar group is a means to specify multiple calendars; the calendar group specified is not saved.

Element 1: Group Name

group-name: Specify the local calendar group to be saved.

Element 2: User ID

user-ID: Specify the user ID of the owner of the calendar group to be saved.

Element 3: User Address

user-address: Specify the user address of the owner of the calendar group to be saved.

DEV

Specifies the name of the device to which the calendar is saved.

***SAVF:** The calendar is saved to a save file.

Element 1: Diskette Device

diskette-device-name: Specify the name of the diskette device on which the calendar is saved. Up to 4 diskette device names can be specified.

Element 2: Tape Device

SAVCAL

tape-device-name: Specify the name of the tape device on which the calendar is saved. Up to 4 tape device names can be specified.

DATE

Specifies a range of dates. All items on the calendar between the dates, including items that occur on the start date and end date, are saved.

Note: If *AVAIL is specified for the start date and *END for the end date, all items on the calendar are saved.

Element 1: Start Date

***AVAIL:** All calendars on or before the end date specified are saved.

start-date: Specify a start date.

Element 2: End Date

***END:** All calendars on or after the start date specified are saved.

end-date: Specify an end date.

RMVITM

Specifies whether calendar items are deleted from the calendar when they are saved.

***NO:** Calendar items are not deleted from the calendar when they are saved.

***YES:** Calendar items are deleted from the calendar when they are saved.

VOL

Specifies the volume identifiers of the volumes on which the data is saved. The volumes must be placed in the device in the order specified on this parameter. More information on this parameter can be found in Appendix A of the *CL Reference* manual.

***MOUNTED:** The volume identifier of the tape or diskette placed on the device is used.

volume-identifier: Specify the volume identifier. A maximum of 50 volume identifiers can be specified.

SEQNBR

Specifies, for a tape device, the sequence number of the file into which the calendars are saved.

***END:** The calendars are saved on the end of the tape.

sequence-number: Specify a sequence number.

EXPDATE

Specifies the expiration date of the save files. The save files cannot be overwritten until the specified expiration date.

***PERM:** The save files are protected permanently.

expiration-date: Specify an expiration date.

ENDOPT

Specifies, for a tape device, whether the tape is rewound, left, or rewound and unloaded after the calendar items are saved.

***REWIND:** The tape is rewound.

***LEAVE:** The tape is left where it is.

***UNLOAD:** The tape is rewound and unloaded.

LABEL

Specifies the label that identifies the data file on the tape or diskette used for the save operation.

***GEN:** The system generates the label name.

data-file-identifier: Specify the data file identifier.

SAVF

Specifies the qualified name of the save file in which the calendar is saved.

The possible library values are:

***LIBL:** The library list is used to locate the file.

***CURLIB:** The current library for the job is used to locate the file. If no library is specified as the current library for the job, the QGPL library is used.

library-name: Specify the name of the library where the file is located.

save-file-name: Specify the name of the save file.

CLEAR

Specifies whether uncleared save files or uncleared tapes and diskettes encountered during the save operation are automatically cleared. An uncleared tape or diskette contains a file whose expiration date has not

passed. This parameter will not initialize the tape or diskette.

***NONE:** None of the uncleared tapes, diskettes, or save files is cleared. If an uncleared tape, diskette, or save file is encountered, an inquiry message is sent.

***ALL:** All uncleared tapes, diskettes, or save files are cleared. If a sequence number is specified when saving the calendar to a tape, the tape is cleared beginning with the file at that sequence number.

***AFTER:** All uncleared tapes or diskettes after the first tape or diskette are cleared. If the first tape or diskette is not cleared, an inquiry message is sent.

DTACPR

Specifies whether data compression is performed.

***DEV:** Data compression is performed if the hardware compression feature is installed on the target tape device. If the compression feature is not installed or the calendar is being saved to a diskette or save file, data compression is not performed.

***NO:** Data compression is not performed.

***YES:** Software data compression is performed. If the calendar is being saved to tape and the hardware compression feature is installed on the target device, hardware data compression is performed.

CMDCHRID

Specifies the graphic character set and code page values for the data entered as command parameter values.

Note: CMDCHRID(*SYSVAL) cannot be specified if this command is used interactively from a device with a CHRID value other than *SYSVAL.

***SYSVAL:** The system value, QCHRID, is used.

***DEV:** The character set and code page defined for the device is used. This value cannot be used when running the command in batch.

Element 1: Character Set

character-set: Specify the character set. Valid values range from 1 through 9999.

Element 2: Code Page

code-page: Specify the code page. Valid values range from 1 through 9999.

TGTRLS

Specifies the release level of the operating system on which you intend to restore and use the calendar object.

You can specify an exact release level in the format VxRxMx, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V2R3M0 is version 2, release 3, modification level 0. The possible values are:

***CURRENT:** The calendar object created or saved on the current release will restore and function correctly on the current release or on any future release. For example, any object created or saved using *CURRENT will restore and function only on V2R3M0 or any future release.

***PRV:** The calendar object created or saved on the current release will restore and function correctly on the previous release or any later release. For example, any object created or saved using *PRV will restore and function only on V2R2M0 or any later release.

release-level: Specify the release level in the format VxRxMx. The object can be saved to a system with the specified release or with any subsequent release of the operating system installed.

Note: Valid values change for each release level. The valid TGTRLS values for this release are:

- A value of V2R2M0. This means that an object created or saved on the current release will restore and function correctly on V2R2M0 or any later release. For instance, an object created or saved using TGTRLS(V2R2M0) will restore and function on V2R2M0 or any later release. This value is the same as *PRV.
- A value of V2R3M0. This means that an object created or saved on the current release will restore and function correctly on V2R3M0. For instance, an object created or saved using TGTRLS(V2R3M0)

SAVCAL

| will function on V2R3M0 or any
| future release. This value is the
| same as *CURRENT.

Examples

Example 1: Saving a Scheduling Calendar

```
SAVCAL CAL(*SCHED) DEV(TAP01)
```

This command saves the scheduling calendar for the user issuing the command to the device named TAP01.

Example 2: Saving and Deleting Calendar Items

```
SAVCAL CAL(*ALL *ALL) DATE(*AVAIL 010190)  
      RMVITM(*YES) DEV(TAP01)
```

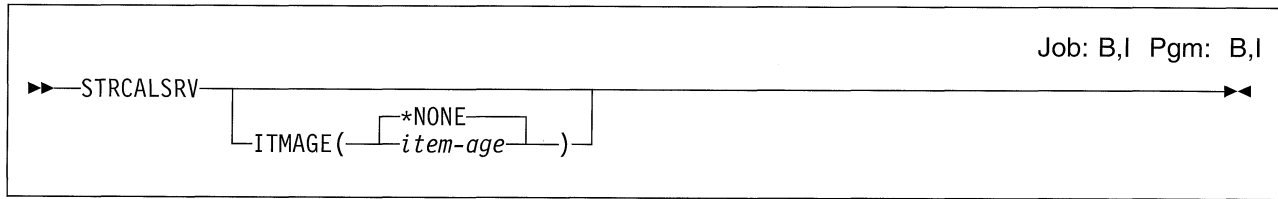
This command saves and then deletes the items on or before January 1, 1990, from all the calendars on the system. The calendars are saved to the device names TAP01. The user must be a security officer to issue this command.

Example 3: Saving and Printing Calendars

```
SAVCAL CAL(*SCHED DEPT54P DSTL) DEV(*SAVF)  
      SAVF(RMSLIB/CALSAVF)
```

This command saves the scheduling calendars for the users listed in the DEPT54P DSTL distribution list. The calendars are saved to a save file named CALSAVF in library RMSLIB.

STRCALSRV (Start Calendar Service) Command



Purpose

The Start Calendar Service (STRCALSRV) command starts the calendar services environment (QCALSrv) in the system work subsystem (QSYSWRK).

Parameters

ITMAGE

Specifies the age (in number of days) of an item to be skipped.

Any item with a reminder message on or before the current date minus the age in days specified on this parameter is skipped by the

calendar service. The status of the item is set as though it had been processed but it is not processed again by calendar service.

***NONE:** No item is skipped.

item-age: Specify a the age (in number of days) of the item to be skipped. Valid values range from 1 through 999.

Example

```
STRCALSRV ITMAGE(1)
```

This command starts the calendar service environment in the system work subsystem. Any item with a reminder message date on or before yesterday is skipped.

STRIDXMOM (Start Index Monitor) Command

```
Job: B,I Pgm: B,I REXX: B,I Exec  
▶▶STRIDXMOM◀◀
```

Purpose

The Start Index Monitor (STRIDXMOM) command starts the index monitor. Automatic index updates and reorganizations take place only if the index monitor is started.

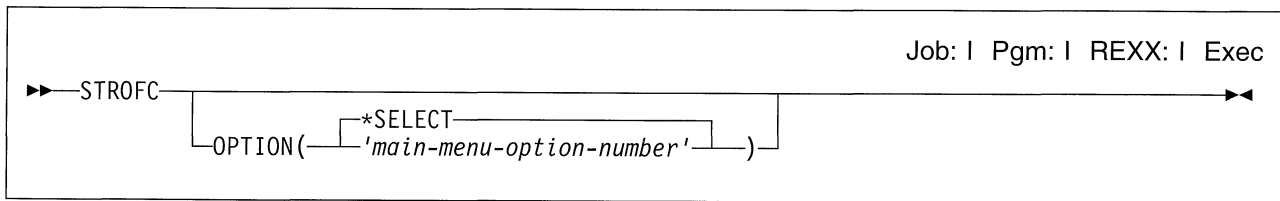
There are no parameters for this command.

Example

```
STRIDXMOM
```

This command starts the index monitor.

STROFC (Start Office) Command



Purpose

The Start Office (STROFC) command allows the user to use the office functions provided by the OfficeVision/400.

Restrictions: The user must be enrolled in the system distribution directory and in the OfficeVision/400 to use this command. The user must also have the OfficeVision/400 licensed program installed.

Parameters

OPTION

Specifies a number that corresponds to an option on the OfficeVision/400 main menu.

***SELECT:** The main menu for OfficeVision/400 is shown. The OfficeVision/400 attention key handling is

called after any option is selected. An attention key event defined before using the OfficeVision/400 is suspended for the remainder of the session.

'main-menu-option-number': Specify a number, ranging between 1 through 9, or 50, that corresponds to a main menu option. (If this parameter value is selected, the main menu display is not shown; the display represented by the menu option number is shown instead.) The OfficeVision/400 attention key handling is not called. An attention key event defined before using the OfficeVision/400 remains active.

Example

```
STROFC OPTION(*SELECT)
```

This command starts the OfficeVision/400 support and shows the OfficeVision/400 main menu.

STRPRSDIR (Start Personal Directory) Command

| | |
|-------------------------|--|
| ▶▶—STRPRSDIR— | Job: B,I Pgm: B,I REXX: B,I Exec ◀◀ |
|-------------------------|--|

Purpose

| The Start Personal Directory (STRPRSDIR) command starts the OfficeVision/400 personal directory utility. The Specify Personal Directory Criteria display is shown when this command is used. From this display, you can create, change, delete, or search personal directories.

| **Restriction:** You must have *USE authority to use this command.

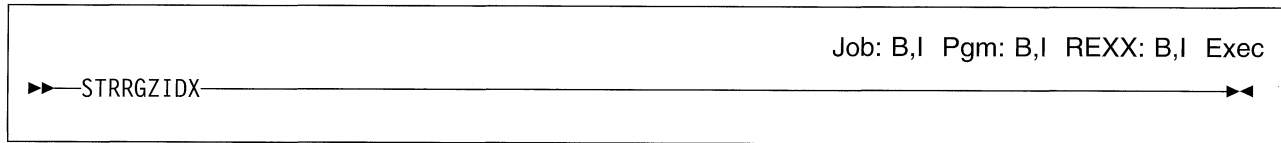
| There are no parameters for this command.

Example

| STRPRSDIR

| This command starts the OfficeVision/400 personal directory function and shows the Specify Personal Directory Criteria display.

STRRGZIDX (Start Reorganization of Index) Command



Purpose

The Start Reorganization of Index (STRRGZIDX) command starts a reorganization of the text index.

The update operation indexes documents on the scheduling queue. The index entries created by the update are temporarily placed in the additional index. The reorganization moves the index entries from the additional index to the primary index.

Running a reorganization does not stop automatic reorganizations.

There are no parameters for this command.

Example

STRRGZIDX

This command starts a reorganization of the text index.

STRUPDIDX (Start Update of Index) Command

| |
|----------------------------------|
| Job: B,I Pgm: B,I REXX: B,I Exec |
| ▶▶—STRUPDIDX—▶▶ |

Purpose

The Start Update of Index (STRUPDIDX) command starts a text index update.

The update operation indexes documents in the scheduling queue. The index entries created by the update are temporarily placed in the additional index. The index entries in the additional index are added to the primary index during a reorganization operation. Use the Start Reorganization of

Index (STRRGZIDX) command to request an index reorganization.

Running an update does not stop automatic index updates.

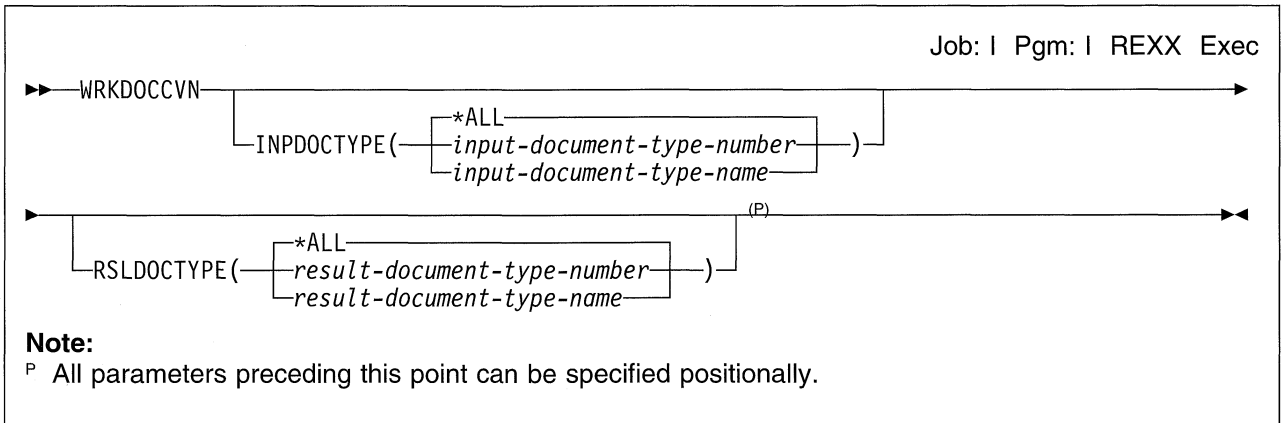
There are no parameters for this command.

Example

```
STRUPDIDX
```

This command requests the start of a text index update.

WRKDOCCVN (Work with Document Conversions) Command



Purpose

The Work with Document Conversions (WRKDOCCVN) command allows the user to show the Work with Document Conversions display. From this display, the user can choose to work with a specific conversion, all conversions, or all conversions to and from a specific document type.

Restriction: The user of this command must have security officer (*SECOFR) authority or be enrolled in OfficeVision/400 as an administrator.

Parameters

INPDOCTYPE

Specifies the name or number of the document type used as input for the document conversion definitions to be shown.

***ALL:** Document conversions of all input types are shown.

input-document-type-number: Specify the numeric value associated with the Document Interchange Architecture (DIA) document type

requested. Valid values range from 1 through 65535.

input-document-type-name: Specify the 10-character name associated with the defined document type.

RSLDOCTYPE

Specifies the name or number of the document type used as the result type for the document conversion definitions to be shown.

***ALL:** Document conversions of all result types are shown.

result-document-type-number: Specify the numeric value associated with DIA the document type requested. Valid values range from 1 through 65535.

result-document-type-name: Specify the 10-character name associated with the defined document type.

Example

```
WRKDOCCVN RSLDOCTYPE(RFTDCA)
```

This command allows the user to work with all conversions that produce a document type of RFTDCA.

WRKDOCTYP (Work with Document Types) Command

| | |
|--|-------------------------|
| | Job: Pgm: REXX Exec |
| <pre> ▶▶ WRKDOCTYP SET ([*ALL *NOSET]) (P) set-ID </pre> | ▶▶ |
| <p>Note: P All parameters preceding this point can be specified positionally.</p> | |

Purpose

The Work with Document Type (WRKDOCTYP) command allows the user to display and work with a subset of the document types defined on the system.

Parameters

SET

Restricts the types to work with a particular set.

***ALL:** There will be no restrictions on the types worked with. All types on the system will appear.

***NOSET:** The types worked with will be restricted to those types that are members of *NOSET.

set-ID: The types worked with will be restricted to those that are a member of the set identified by set-ID.

WRKOFCAPPD (Work with Office Application Descriptions) Command

Job: B,I Pgm: B,I REXX Exec

▶▶ WRKOFCAPPD ◀◀

Purpose

- | The Work with Office Application Description (WRKOFCAPPD) command allows the user to display and perform specific operations on the office applications defined on the system.
- | There are no parameters for this command.

Example

- | WRKOFCAPPD
- | This command allows the user to display and work with all the office applications defined on the system.

WRKOFCAPE (Work with Office Application Entries) Command

Job: | Pgm: | REXX Exec

```

  >> WRKOFCAPE—USRPRF ( — [ *CURRENT —
                          [ *SYSTEM —
                          [ user-profile-name — ] ] ] ) — (P) — >>
  
```

Note:
 P All parameters preceding this point can be specified positionally.

Purpose

The Work with Office Application Entries (WRKOFCAPE) command allows the user to manage the applications associated with sets for a user.

Parameters

USRPRF

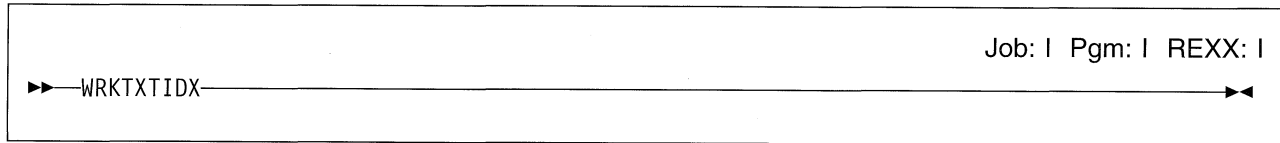
The user whose set-application associations are to be worked with.

***CURRENT:** The current user signed on.

***SYSTEM:** The default values to be used if there is no explicit application for a set.

user-profile-name: The user profile whose association will be worked with.

WRKTXIDX (Work with Text Index) Command



Purpose

The Work with Text Index (WRKTXIDX) command shows the Work with Text Index display from which you can do the following:

- Start update and reorganization of the text index.
- Start and stop the index monitor.
- View the current status of the text index.
- View the message queue.
- Hold or release the text index.
- Gain access to displays.

The displays you can access with the WRKTXIDX command are as follows:

- The Change Text Index Details display is used for setting up or changing the conditions of updates and reorganization.
- The Display Text Index Details display is used for viewing the conditions for automatic updates and reorganization.
- The Display All Text Index Status display is used for viewing status information about the text index.

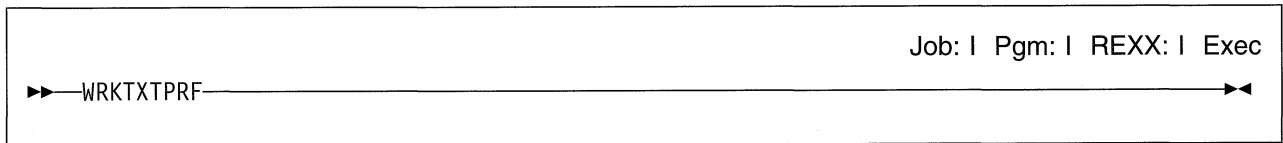
There are no parameters for this command.

Example

WRKTXIDX

This command shows the Work with Text Index display.

WRKTXTPRF (Work with Text Profiles) Command



Purpose

The Work with Text Profiles (WRKTXTPRF) command allows you to work with the Work with Text Profiles display. From this display, you can create, revise, copy, or delete profiles that provide default values when using the word processing function of OfficeVision/400

More information on text profiles is in *Using OfficeVision/400* Word Processing*.

There are no parameters for this command.

Example

```
WRKTXTPRF
```

This command allows you to use the Work with Text Profiles display.

Appendix E. Hierarchical File System (HFS) and DLS

The Document Library Services (DLS) file system is the file system used with OfficeVision/400. You can also use the Hierarchical File System (HFS) APIs to work with DLS documents and folders. This appendix describes the parts of the document library system (DLS) that are especially relevant to applications. It covers these topics:

- Terms and concepts that differ from those used in other hierarchical file systems
- User enrollment
- Authorities needed to work with DLS objects
- Directory entry attributes
- HFS Control File System API (QHFCTLFS)

Before working with the DLS file system through HFS, review these publications:

- *System Programmer's Interface Reference* describes the generic form of the HFS APIs.
- *Interchange Document Profile Reference* provides information for working with attributes.

Terms and Concepts

The primary components of the DLS file system are documents and folders. However HFS and DLS use different terms to refer to these objects:

Figure E-1. DLS Terminology

| HFS Term | DLS Term |
|---------------------------|--|
| File | Document A DLS document can contain up to 2 147 483 647 bytes. DLS document (*DOC) objects differ from OS/400 file (*FILE) objects. The former are byte-stream files like those in other hierarchical file systems. The latter are system database files. |
| Directory | Folder A DLS folder can contain up to 65 535 folders and documents. As the number of objects in a folder increases, the time to access the objects also increases, especially when deleting and inserting objects. For best performance, limit folders to 1000 objects. |
| File or directory | Document library object (DLO). This is a collective term for documents and folders in the DLS file system. |
| Directory entry attribute | Interchange document profile (IDP) parameter. |

The following sections describe DLS file (document) and directory (folder) naming conventions, file types, and the ways in which users can manipulate DLS document and folders using the HFS APIs.

File and Directory Names

See Chapter 5, “Document Library Services” on page 5-1 for a complete discussion of the rules for specifying DLS file (document) and directory (folder) names. In general, path names for the DLS file system have the same format as path names for ordinary hierarchical file systems, described in the *System Programmer’s Interface Reference*.

The total length of a DLS path name cannot exceed 82 characters, including the file system name (QDLS) and all leading and separating slashes (/). If the path name specifies a directory rather than a file, it cannot exceed 69 characters.

File Types

DLS file system users can create files (documents) with data of different types. Some types, such as those within Document Content Architecture, are registered with Document Interchange Architecture (DIA). System users must be careful not to alter these types. For example, if a user creates a document of a type that other users expect to be in DIA form but that does not have DIA form, unpredictable results can occur. Users might not be able to edit or otherwise process the file any more. For further details about DLS document types, see the DOCTYPE parameter of the File Document (FILDOC) command in the *CL Reference*.

Manipulating Files and Directories

Users can manipulate DLS files (documents) and directories (folders) in several ways. For example, a file (document) created with the Open Stream File (QHFOFNSF) API can be deleted later with the Delete Document Library Object (DLTDLO) command. A directory (folder) can be created with the Create Folder (CRTFLR) command, and that folder can be manipulated with directory APIs. Chapter 5, “Document Library Services” on page 5-1 and Chapter 6, “Security Services” on page 6-1 lists the CL commands you can use to work with DLS file system objects. Users can also work with documents and folders through the PC Support/400 and OfficeVision/400 licensed programs.

User Enrollment

Before users can work with the DLS file system, they must be enrolled in the system distribution directory. To enroll and remove users, use the OfficeVision/400 administration option, described in *Managing OfficeVision/400**, or the Add Directory Entry (ADDDIRE), Work with Directory (WRKDIR), and Remove Directory Entry (RMVDIRE) commands, described in the *CL Reference* or in Chapter 4, “Document Distribution Services” on page 4-1.

Object Authorities

The DLS file system uses an object authority system that closely resembles the regular AS/400 authority system. See Chapter 6, “Security Services” on page 6-1 for a detailed discussion of object authority.

Directory Entry Attributes

The following sections discuss these topics about directory entry attributes:

- Standard attributes
- Interchange document profile (IDP) attributes
- Attributes you can use when creating directories and files
- Retrieving attributes

Standard Attributes

You can use these standard directory entry attributes for DLS directories and files:

- QALCSIZE
- QCRTDTTM
- QFILATTR
- QFILSIZE
- QNAME
- QWRDTTM

Note: DLS does not support the standard attribute QACCDTTM. You can specify it in an attribute list, but it will be ignored.

These attributes are described in detail in the *System Programmer's Interface Reference*.

If you misspell a standard attribute name, the DLS file system handles the attribute as an extended attribute, which in most cases would not produce meaningful results.

Interchange Document Profile (IDP) Attributes

Except for the standard attributes, all directory entry attributes for DLS files and directories are stored in the interchange document profile (IDP). The IDP is defined by the Document Interchange Architecture (DIA) and is a part of each DLS object that stores information about the object. The IDP is separate from the object's data and is divided into logical parts called subprofiles. The IDP contains both extended attributes, which are not defined by DIA, and DIA-defined attributes. Directory entry attributes stored in the IDP are sometimes called **IDP parameters**. For more information about the IDP and DIA, see the *Interchange Document Profile Reference*.

Attribute Names

When specifying the names of attributes stored in the IDP in calls to HFS APIs, applications must use the format used by the IDP. Attributes are stored in the IDP as fields. An IDP field can contain a DIA-defined attribute or an extended attribute. All DIA-defined IDP fields can be accessed by specifying a name in the form DIA.sssspppp, where ssss is the hexadecimal character representation of the subprofile identification and pppp is the hexadecimal character representation of the code point of the field as it is identified in the IDP. For example, the base subprofile has an ID of hex CA04. The author field in the base subprofile has a code point of hex C704. You can retrieve the author field by requesting the attribute named DIA.CA04C704.

Attribute names in this form must be exactly 12 characters long.

The profile graphic code-page identifier (GCID) has a code point of hex C701 and is also in the base subprofile. You can retrieve it by requesting the attribute named DIA.CA04C701. The profile GCID is set at file creation time and it cannot be updated.

Any attribute name that has the letters DIA and a period (DIA.) as the first 4 characters is treated by the DLS system as a DIA attribute name and must be in the form described here and in the *Interchange Document Profile Reference*. If the name and data are not specified in this form, an error is returned.

Some standard attributes can also be accessed using this form of specification. When this is done, the data returned might be in a different form from the data returned when the attribute is specified as a standard attribute. For example, the date and time attributes are returned in DIA internal form, not in the form specified for standard attributes.

Attribute Values

When applications pass attribute values to or receive attribute values from the DLS system in the attribute information and selection tables, the values are in the following form:

The first field is two bytes long and is the code point of the attribute (CP).

The second field is one byte long and is the format type (F). It can be either X'01' (a single-parameter structure) or X'41' (a parameter-set structure).

For example, for the author field (DIA.CA04C704) either of the following forms could apply:

1. Single parameter structure (format 01). This form could be used when the only subfield that is defined is the author name:

```
      CP  F
|x'C704'|x'01'|author|
```

2. Parameter set structure (format 41). This form could be used when a profile GCID and other data are associated with the author name:

```
      CP    F    L    T
|x'C704'|x'41'|x'06'|x'01'|gcid|
```

```
      L    T
x'04'|x'14'|lgrpid|
```

```
      L    T
x'nn'|x'15'|ctryid|
```

```
      L    T
x'nn'|x'16'|nlgid|
```

```
      L    T    L    T
x'02'|x'05'|x'nn'|x'02'|author|
```

```
      L    T
x'nn'|x'03'|dgn|
```

```
      L    T
x'nn'|x'04'|den|...
```


The length field (L) contains the length of the length field and the next type and data fields. In this example, the first length field includes the length itself (1 byte), the type (1 byte), and the GCID (4 bytes) for a total of 6 bytes.

The type field (T) contains the type of data that follows. The meaning of each type can be unique for each code point.

Not all of the fields shown are required, and not all available fields are shown. In this parameter the fields can be repeated to describe more than one author. Refer to the *Interchange Document Profile* for complete information.

For the profile GCID, the following form (format 01) is the only acceptable form:

```
CP F
|x'C701'|x'01'|gcid|
```

When using the parameter set structure (format 41), the presence or absence of any given field is not assured. The data returned is exactly what is stored in the profile and can vary. For example, there might be one, more than one, or no explicit GCIDs for the fields. If a field requires a GCID but does not have an explicit GCID, it uses the profile GCID.

Attributes with Multiple Values

When an IDP attribute is updated, the entire attribute is replaced. When you are using the HFS APIs, you cannot add or replace one value of a multiple-value attribute. For example, you cannot add or remove an author to an existing list of authors, but you can replace the entire list: retrieve the author attribute, change the attribute contents, and replace the attribute. The *Interchange Document Profile Reference* describes the format of all profile parameters.

Extended Attributes

Any attribute that is not a standard attribute and whose name does not start with DIA., is an extended attribute. Extended attributes are stored in the extended attribute subprofile of the IDP. You can request an extended attribute by name, like any other attribute. The name can be any character string up to 128 characters long that does not begin with the characters DIA. (the letters DIA plus a period). In addition, to avoid confusion with standard attributes, you should not give extended attributes names beginning with the letter Q. The data can be in any form chosen by the user.

Attributes to Use When Creating Directories and Files

When creating a directory (folder) in the DLS file system, you can specify only two attributes: text description and profile GCID. When creating or replacing a file (document), you can specify only these attributes:

- Text description (DIA.CA04C700)
- Profile GCID (DIA.CA04C701)
- File type (DIA.CA04C706)
- QFILSIZE
- QFILATTR

To set other attributes, use the QHFCHGAT (Change Directory Entry Attributes) API after the file or directory is created. For more information on QHFCHGAT, see the *System Programmer's Interface Reference*.

File type

(The file's document type) This attribute can have many different values; for a list, see the *Interchange Document Profile Reference*. The attribute name is DIA.CA04C706. The attribute data must be in the following form, where *type* is the 2-byte value of the document type desired:

| | | |
|---------|-------|------|
| x'C706' | x'01' | type |
|---------|-------|------|

The length-of-attribute-value field specified in the API's attribute information table must be 5. If this attribute is not provided when a file is opened, a value of hex 000E is assumed. This value is the document type for an IBM Personal Computer file.

Profile GCID

(Profile graphic code-page identifier) The GCID is a code that identifies which graphic elements, such as letters and numbers, are associated with any value in a character. The GCID is usually associated with a language; sometimes each country that uses a given language uses a separate GCID. The GCID identifies the code page and character set in which the data is stored and can be used in translating text data from one code page to another. Sometimes called the character ID (CHRID) instead, the GCID is also used to translate keyboard input into a common code page when creating AS/400 display files.

The profile GCID identifies the code page of the attribute data, for attributes that do not have an explicit GCID associated with them. You can specify the profile GCID attribute when creating or replacing a file and when creating a directory. The attribute name is DIA.CA04C701. The attribute data must be in the following format, where *gcid* is the 4-byte value of the graphic code-page identifier for the profile:

| | | |
|---------|-------|------|
| x'C701' | x'01' | gcid |
|---------|-------|------|

The length-of-attribute-value field specified in the API's attribute information table must be 7. If this attribute is not provided, a profile GCID of code page 697 and character set 500 is used.

You cannot change this attribute with the Change Directory Entry Attributes (QHFCHGAT) API.

Text description

An internal document name 1 to 44 characters long. The attribute name is DIA.CA04C700. The attribute data must be in the following format, where *text* is the character data describing the directory entry:

| | | |
|---------|-------|------|
| x'C700' | x'01' | text |
|---------|-------|------|

The length-of-attribute-value field specified in the attribute information table must be the length of the text string, plus 3 for the fields preceding the text. The text must be encoded in the character set and code page specified by the GCID.

If you do not specify this field when creating a file or directory, the file system uses the name of the file or directory being created as the value.

You can change this attribute with the Change Directory Entry Attributes (QHFCHGAT) API, but you cannot delete it.

Retrieving Attributes

When you retrieve directory entry attributes in the DLS file system using the Read Directory Entries (QHFRDDR) API, the returned data might appear to be incomplete or incorrect. This is because of the manner in which the DLS file system performs open, read, and delete operations. When a user opens a DLS directory (folder), the system records the folders and files (documents) that are in that directory at that time. After the open operation:

- If more directories or files are added to the directory before the read operation, they do not appear in the data returned to the user.
- If files or directories have been deleted from the directory, they might or might not appear in the output from the read operation:
 - If the user requests only standard directory entry attributes, the deleted objects might appear in the output.
 - If the user requests additional attributes that must be extracted from the object, the deleted objects are ignored and no error is returned to the user.

HFS Control File System API (QHCTLFS)

DLS supports the HFS Control File System (QHCTLFS) API. See the *System Programmer's Interface Reference* manual for more information on the QHCTLFS API.

The following format of the QHCTLFS API is unique to the DLS file system.

Note: The file system name, not a file handle, must be specified (the second parameter to QHCTLFS must be QDLS). For complete information on the QHCTLFS API see the *System Programmer's Interface Reference* manual.

| QHCTLFS Input Data Buffer | | |
|---------------------------------|---|-----------|
| Function to perform | INPUT | BINARY(4) |
| Only one function is supported: | | |
| 1 | Suspend file system. This will close all internal DLS database files and end DLS commitment control. This function will fail if any files or directories are open, and will return information about all open files and directories in the output buffer. | |

| QHCTLFS Output Data Buffer | | |
|--|--------|-----------|
| Count of open files and directories | OUTPUT | BINARY(4) |
| The number of open files and directories. | | |
| <i>Repeated for each open file or directory</i> | | |
| Offset to file or directory information | OUTPUT | BINARY(4) |
| The offset to the information for an open file or directory. | | |
| <i>Repeated for each offset (open file or directory)</i> | | |

| QHCTLFS Output Data Buffer | | |
|---|---------------|------------------|
| Length of path and name | OUTPUT | BINARY(4) |
| The length of the combined directory path and name of the open file or directory. | | |
| File or directory indicator | OUTPUT | CHAR(1) |
| The file or directory indicator. F This entry is for an open file D This entry is for an open directory | | |
| Path and name of file or directory | OUTPUT | CHAR(*) |
| The directory path and name of the open file or directory. | | |

Bibliography

The following manuals contain information you may need. The manuals are listed with their full title and base order number. When these manuals are referred to in this guide, the short title listed is used.

The following AS/400 manuals contain related information you may find useful:

- *Application Development Tools: Screen Design Aid User's Guide and Reference*, SC09-1340, explains how to use the screen design aid (SDA) to design, create, and maintain displays, menus, and online help information.

Short Title: *SDA User's Guide and Reference.*

- *Basic Backup and Recovery Guide*, SC41-0036, provides information about the different media available to save and restore system data. This manual describes how to record changes made to database files and how that information can be used for system recovery and activity report information. It also provides information on how to install the system from a backup.

Short Title: *Basic Backup and Recovery Guide.*

- *Business Graphics Utility User's Guide and Reference*, SC09-1408, provides information about using the AS/400 Business Graphics Utility (BGU) to create various types of charts.

Short Title: *BGU User's Guide and Reference.*

- *Guide to Enabling C2 Security*, SC41-0103, describes how to customize your system to meet the requirements for C2 Security, as described in the *Department of Defense Trusted Computer Evaluation Criteria*.

Short Title: *Guide to Enabling C2 Security.*

- *Communications: Advanced Peer-to-Peer Networking Guide*, SC41-8188, provides information about advanced peer-to-peer networking (APPN) support provided by the AS/400 system. It describes concepts, functions, and features as carried out on the AS/400 system.

Short Title: *APPN Guide.*

- *Communications: Advanced Program-to-Program Communications Programmer's Guide*, SC41-8189, provides information about the advanced program-to-program communications (APPC) support provided by the AS/400 system. This guide includes application program considerations, configuration requirements and commands, problem management for APPC, and general networking considerations.

Short Title: *APPC Programmer's Guide.*

- *Communications: Distribution Services Network Guide*, SC41-9588, provides information about configuring a network for Systems Network Architecture distribution services (SNADS) and the Virtual Machine/Multiple Virtual Storage (VM/MVS) bridge. This guide also discusses object distribution functions, document library services, and system distribution directory services.

Short Title: *Distribution Services Network Guide.*

- *Communications: Operating System/400* Communications Configuration Reference*, SC41-0001, provides general OS/400 configuration information, including detailed descriptions of network interface, line, controller, device, mode, and class-of-service descriptions, configuration lists, and connection lists.

Short Title: *OS/400* Communications Configuration Reference.*

- *Data Description Specifications Reference*, SC41-9620, provides detailed description of the entries and keywords needed to describe database files (both logical and physical) and certain device files (for displays, printers, and intersystem communications function (ICF)) external to the user's programs.

Short Title: *DDS Reference.*

- *Data Management Guide*, SC41-9658, provides information to help the programmer manage key aspects of the system. For example, it describes how to use database files.

Short Title: *Data Management Guide.*

- *Database Guide*, SC41-9659, provides information about using files in application programs. This manual contains information regarding fundamental structure and concepts of data management support, use of system commands to copy data from one place to another, and instructions on tailoring a system using double-byte data.

Short Title: *Database Guide.*

- *Device Configuration Guide*, SC41-8106, provides information on how to do an initial local hardware configuration and how to change that configuration. It also contains conceptual information about device configuration and planning information for device configuration.

Short Title: *Device Configuration Guide.*

- *National Language Support Planning Guide*, GC41-9877, provides information required to understand and use the national language support function on the AS/400 system. This guide helps in preparing the AS/400 user for planning, installing,

configuring, and using the AS/400 National Language Support (NLS) and multilingual system. It also provides an explanation of the database management of multilingual data and application considerations for a multilingual system.

Short Title: *National Language Support Planning Guide.*

- *PC Support/400: DOS Installation and Administration Guide*, SC41-0006, provides information needed to set up, configure, and tailor PC Support/400 through batch and configuration files, using the DOS program.

Short Title: *PC Support/400 DOS Installation and Administration Guide.*

- *PC Support/400: DOS User's Guide*, SC41-8199, provides concepts and examples of how to use the PC Support/400 functions with the DOS program. It assumes that the PC Support/400 product already installed and set up.

Short Title: *PC Support/400 User's Guide for DOS.*

- *PC Support/400: OS/2 Installation and Administration Guide*, SC41-0007, provides information for planning and installing PC Support/400, and configuring and diagnosing problems for individual PC Support/400 users. This manual is intended for users with personal computers using the OS/2 operating system

Short Title: *PC Support/400 OS/2 Installation and Administration Guide.*

- *PC Support/400: OS/2 User's Guide*, SC41-8200, provides concepts and examples of how to use the PC Support/400 functions with the OS/2 program. It assumes that the PC Support/400 product is already installed and set up.

Short Title: *PC Support/400 User's Guide for OS/2.*

- *Programming: Control Language Programmer's Guide*, SC41-8077, provides a wide-ranging discussion of AS/400 programming topics, including:
 - A general discussion of 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
 - How to define and create user-defined commands and menus
 - Application testing, including debug mode, breakpoints, traces, and display functions

Short Title: *CL Programmer's Guide.*

- *Programming: Control Language Reference*, SC41-0030, describes the AS/400 control language and its commands. This manual is divided into several volumes.

- The first volume provides an overview of all the CL commands for the AS/400 system, and it describes the syntax rules needed to code them.
- The other volumes describe all the OS/400 commands, which are arranged alphabetically by command name. Each CL command description includes a syntax diagram, descriptions of its parameters, default values, keywords, and examples.

Short Title: *CL Reference.*

- *Programming: GDDM Programming Guide*, SC41-0536, provides information about using OS/400 graphical data display manager (GDDM) to write graphics application programs.

Short Title: *GDDM Programming Guide.*

- *Programming: GDDM Programming Reference*, SC41-0537, provides information about using OS/400 graphical data display manager (GDDM) to write graphics application programs. This manual provides detailed descriptions of all graphics routines available in GDDM. It also provides information about high-level language interfaces to GDDM.

Short Title: *GDDM Programming Reference.*

- *Programming: Reference Summary*, SX41-0028, provides programmers quick and easy access to summary information about many of the languages and utilities available on the AS/400 system. This information includes:
 - All AS/400 CL commands in various forms
 - Information related to CL commands and the IBM-supplied files that are used by some commands
 - IBM-supplied objects, including libraries
 - IBM-supplied system values

Short Title: *Programming Reference Summary.*

- *Programming: Work Management Guide*, SC41-8078, provides information about how to create and change a work management environment. Other topics include a description of tuning the system, collecting performance data, working with system values to control or change the overall operation of the system, and a description of how to gather data to determine who is using the system and what resources are being used.

Short Title: *Work Management Guide.*

- *Query/400 User's Guide*, SC41-9614, provides detailed information about how to use AS/400

Query to get data from any database file. It describes how to sign on Query, and how to define and run queries to create reports containing the selected data. This manual describes all the tasks for defining the query of one or more files, the way the report is to look when the query is run, and whether the report is to be displayed, printed, or put in a database file.

Short Title: *Query/400 User's Guide.*

- *Security Reference*, SC41-8083, provides programmers with information about AS/400 system security and security issues within OfficeVision/400.

Short Title: *Security Reference.*

- *System Programmer's Interface Reference*, SC41-8223, provides information on how to create, use, and delete objects that help manage system performance, use spooling efficiently, and maintain database files efficiently. This manual also includes information on creating and maintaining the programs for system objects and retrieving OS/400 information by working with objects, database files, jobs, and spooling.

Short Title: *System Programmer's Interface Reference.*

- *Systems Application Architecture* OfficeVision/400*: Managing OfficeVision/400*, SC41-9627, provides information on how to manage the day-to-day activities of OfficeVision/400. It also includes information on maintaining office enrollment and creating and managing office objects.

Short Title: *Managing OfficeVision/400*.*

- *Systems Application Architecture* OfficeVision/400*: Planning For and Setting Up OfficeVision/400*, SC41-9626, provides information about planning for and setting up OfficeVision/400. The information includes planning for enrolling users, word processing, mail and calendar processing, using OfficeVision/400 with IBM personal computers, and using OfficeVision/400 in a communications network.

Short Title: *Planning For and Setting Up OfficeVision/400*.*

- *Systems Application Architecture* OfficeVision/400*: Using OfficeVision/400*, SC41-9616, provides infor-

mation on how to use OfficeVision/400, including information on handling mail and calendars.

Short Title: *Using OfficeVision/400*.*

- *Systems Application Architecture* OfficeVision/400*: Using OfficeVision/400 Adapted Word Processing Function*, SC41-9879, provides information on how to do word processing tasks with the adapted word processing function of OfficeVision/400.

Short Title: *Using OfficeVision/400* Adapted Word Processing Function.*

- *Systems Application Architecture* OfficeVision/400*: Using OfficeVision/400 Word Processing*, SC41-9618, provides the office user with information on how to use the word processing functions of OfficeVision/400.

Short Title: *Using OfficeVision/400* Word Processing.*

The following manuals contain information about text data streams:

- *Document Content Architecture: Final-Form Text Reference*, GC23-0757, describes the syntax and semantics for all allowable elements in a text data stream for the interchange of formatted documents.

Short Title: *Final-Form Text Reference.*

- *Document Content Architecture: Revisable-Form Text Reference*, GC23-0758, describes the syntax and semantics for all allowable elements in a text data stream for the interchange of revisable documents.

Short Title: *Revisable-Form Text Reference.*

The following manuals contain information about sending documents across systems:

- *Document Interchange Architecture: Interchange Document Profile Reference*, SC23-0764, provides detailed profile descriptions needed by users who write applications that send documents from system to system.

Short Title: *Interchange Document Profile Reference.*

- *Document Interchange Architecture: Technical Reference*, SC23-0781, describes the system document interchange architecture.

Short Title: *Technical Reference.*

Index

Special Characters

- *ALL authority 6-13
- *ANY 3-2
- *CHANGE authority 6-13
- *DRX 4-5
- *DTO 4-5
- *DUO 4-5
- *EXCLUDE authority 6-13
- *USE authority 6-13

A

access code

- definition 6-18

Add Calendar Item (ADDCALITM) command

- calendar services 2-5
- description D-4
- output file considerations C-1

Add Directory Entry (ADD DIRE) command 3-10

Add Distribution List Entry (ADD DSTLE) command 3-19

Add Document Conversion (ADD DOCCVN) command 11-9, D-12

Add Document Library Object Authority (ADD DLOAUT) command 6-19

Add Office Application Entry (ADD OFCAPPE) command D-14

Add Office Enrollment (ADD OFCENR) command 10-7, D-15

Add Text Index Entry (ADD TXTIDX) command 7-15, D-20

additional index 7-3

address

- definition 3-2

administration monitor 7-2, 7-5

advanced peer-to-peer networking (APPN) definition 10-5

advanced program-to-program communications (APPC)

- considerations for service administration 1-4
- definition 1-2
- use with document distribution services 4-1

*ALL authority 6-13

all object special authority definition 6-5

APPC (advanced program-to-program communications)

- considerations for service administration 1-4
- definition 1-2
- use with document distribution services 4-1

application display file 8-10

application program interface (API)

- QHCTLFS (HFS Control File System) E-7
- QO1PCFID (PCFILE Identification Process) API 11-17
- QOGCHGOE (Change Office Exit Programs) 11-10
- QOGRTVOE (Retrieve Office Exit Programs) 11-11, 11-15
- QOOCTLCL (Control Calendar) 2-28

APPN (advanced peer-to-peer networking) definition 10-5

AS/400 Query

- description 11-5

AS/400 system

- allowing remote users access 10-6
- enrollment 6-3
- functions using document library services 5-3
- PC Support/400 1-13, 11-7
- query 11-5
- SNADS network 4-3
- using with office services 1-2
- using with word processing 8-9

audit

- special authority 6-6

authority

- checking
 - chart 6-16
 - definition 6-15
 - sequence 6-15
- level
 - documents 6-13
 - folders 6-13
 - users 6-4
- parameters 2-4

authorization list management authority 6-8

authorization lists 6-7

B

backup procedures 9-1

BGU (Business Graphics Utility) 11-5

bibliography F-1—F-3

bridge

- VM/MVS 4-3

Business Graphics Utility (BGU) 11-5

C

CAL parameter 2-3

calendar

- definition 2-1
- restoring 9-4

calendar services

- adding item 2-5
- authority 2-1
- changing calendar authority 2-7
- commands 1-6, 2-5
- creating calendar 2-8
- deleting calendar 2-10
- displaying
 - calendar authority 2-11
 - calendar description 2-11
 - calendar item 2-12
- ending 2-12
- group 2-1
- item 2-1
- overview 2-1
- printing calendar 2-12
- querying calendar item 2-13
- removing calendar item 2-15
- restoring calendar 2-16
- saving calendar 2-17
- starting 2-18

calendar user-defined functions

- Control Calendar (QOOCTLCL) API 2-28
- description 2-19
- examples 2-24
- sample program 2-25
- substitution variables 2-20

*CHANGE authority 6-13

Change Calendar Authority (CHGCALAUT)

- command 2-7, D-23

Change Class (CHGCLS) command 4-11

Change Directory Attributes (CHGDIRA)

- command 3-12

Change Directory Entry (CHGDIRE) command 3-12

Change Distribution Description (CHGDSTD)

- command 4-18

Change Document Conversion (CHGDOCCVN)

- command D-28

Change Document Description (CHGDOCD)

- command 5-12, 5-14

Change Document Library Object Auditing Level

- (CHGDLOAUD) command 6-20

Change Document Library Object Authority

- (CHGDLOAUT) command 6-20

Change Document Library Object Owner

- (CHGDLOOWN) command 6-21

Change Document Set (CHGDOCSET)

- command D-30

Change Document Type (CHGDOCTYP)

- command D-31

Change Job (CHGJOB) command 4-11

Change Journal (CHGJRN) command 9-11

Change Office Application Description

- (CHGOFCAPPD) command D-34

Change Office Application Entries (CHGOFCAPPE)

- command D-35

Change Office Attributes (CHGOFCA)

- command 10-3, D-33

Change Office Enrollment (CHGOFCENR)

- command 10-8, D-36

Change Office Exit Programs (QOGCHGOE)

- API 11-10

Change Profile (CHGPRF) command 6-5

Change User Profile (CHGUSRPRF) command

- use with security services 6-5

character identifier (CHRID) parameter 4-13, A-1

character set

- CL command considerations 2-4
- CMDCHRID and CHRID parameter A-1
- common A-1
- considerations A-1
- definition A-1
- document interchange A-6
- document library services 5-11
- override print file (OVRPRTF) command A-4
- print files A-4
- print options display A-3
- QCHRID system value A-4
- supported by word processing A-5
- translating data A-2
- word processing considerations A-3
- work station support A-5

Check Document (CHKDOC) command 8-14, D-42

Check Document Library Object (CHKDLO)

- command 5-12, 5-15

CHRID (character identifier) parameter 4-13, A-1

CL commands

- See commands

- See Control Language (CL) commands

CL programs

- See Control Language (CL) programs

- See example applications

code page

- CL command considerations 2-4
- CMDCHRID and CHRID parameter A-1
- common A-1
- considerations A-1
- definition A-1
- document interchange A-6
- document library services 5-11
- override print file (OVRPRTF) command A-4
- print files A-4
- print options display A-3
- QCHRID system value A-4
- supported by word processing A-5
- translating data A-2
- word processing considerations A-3
- work station support A-5

command interface

- for directory services 3-6
- for document distribution services 4-9
- for word processing 8-6

command interface *(continued)*

office services 1-4
OfficeVision/400 1-1

commands

ADDCALITM (Add Calendar Item) 2-5, D-4
ADDDIRE (Add Directory Entry) 3-10
ADDLLOAUT (Add Document Library Object Authority) 6-19
ADDOCCVN (Add Document Conversion) 11-9, D-12
ADDSTLE (Add Distribution List Entry) 3-19
ADDOFCAPPE (Add Office Application Entry) 11-13, D-14
ADDOFCENR (Add Office Enrollment) 10-7, D-15
ADDTXTIDX (Add Text Index Entry) 7-15, D-20
CFGRPDS (Configure VM/MVS Bridge) 10-1
CHGCALAUT (Change Calendar Authority) 2-7, D-23
CHGCLS (Change Class) 4-11
CHGDIRA (Change Directory Attributes) 3-12
CHGDIRE (Change Directory Entry) 3-12
CHGDLOAUD (Change Document Library Object Auditing Level) 6-20
CHGDLOAUT (Change Document Library Object Authority) 6-20
CHGDLOOWN (Change Document Library Object Owner) 6-21
CHGDOCCVN (Change Document Conversion) D-28
CHGDOCD (Change Document Description) 5-14
CHGDOCSET (Change Document Set) 11-13, D-30
CHGDOCTYP (Change Document Type) 11-13, D-31
CHGDSTD (Change Distribution Description) 4-18
CHGJOB (Change Job) 4-11
CHGJRN (Change Journal) 9-11
CHGOFCA (Change Office Attributes) 10-3, 11-13, D-33
CHGOFCAPPD (Change Office Application Description) 11-13, D-34
CHGOFCAPPE (Change Office Application Entries) 11-13, D-35
CHGOFCENR (Change Office Enrollment) 10-8, D-36
CHGPRF (Change Profile) 6-5
CHGUSRPRF (Change User Profile) 6-5
CHKDLO (Check Document Library Object) 5-15
CHKDOC (Check Document) 8-14, D-42
CPYDOC (Copy Document) 5-17
CPYFRMPCD (Copy From PC Document) 5-20
CPYSPLF (Copy Spooled File) 4-66
CPYTOPCD (Copy To PC Document) 5-21
CRTCAL (Create Calendar) 2-8, D-44
CRTDOC (Create Document) 8-16
CRTDOCSET (Create Document Set) 11-13, D-50
CRTDOCTYP (Create Document Type) 11-13, D-51

commands *(continued)*

CRTDSTL (Create Distribution List) 3-20
CRTFLR (Create Folder) 5-23
CRTOFCAPPD (Create Office Application Description) 11-13, D-54
CRTPF (Create Physical File) 5-20
CRTSPADCT (Create Spelling Aid Dictionary) 7-14, 8-17
CRTSRCPF (Create Source Physical File) 5-20
CVTDOC (Convert Document) 11-13, D-55
DLTCAL (Delete Calendar) 2-10, D-56
DLTDLO (Delete Document Library Object) 5-24
DLTDOCSET (Delete Document Set) 11-13, D-58
DLTDOCTYP (Delete Document Type) 11-13, D-59
DLTDST (Delete Distribution) 4-19
DLTDSTL (Delete Distribution List) 3-22
DLTJRNRCV (Delete Journal Receiver) 9-11
DLTOFCAPPD (Delete Office Application Description) 11-13, D-60
DLTOUTMAIL (Delete Outgoing Mail) 10-3, D-61
DLTSPADCT (Delete Spelling Aid Dictionary) 8-20
DMPDLO (Dump Document Library Object) 5-13
DMPYSOJB (Dump System Object) 4-7
DSPAUTLDO (Display Authorization List Document Library Objects) 6-22
DSPCALAUT (Display Calendar Authority) 2-11, D-62
DSPCALD (Display Calendar Description) 2-11, D-65
DSPCALITM (Display Calendar Item) 2-12, D-68
DSPDIR (Display Directory) 3-13
DSPDLOAUT (Display Document Library Object Auditing Level) 6-23
DSPDLOAUT (Display Document Library Object Authority) 6-23
DSPDLONAM (Display Document Library Object Name) 5-26
DSPDOC (Display Document) 8-22
DSPDOCCVN (Display Document Conversion) D-71
DSPDSTL (Display Distribution List) 3-23
DSPFD (Display File Description) 9-9
DSPFLR (Display Folder) 5-27
DSPHLPDOC (Display Help Document) 8-23
DSPIDXSTS (Display Index Status) 7-16, D-73
DSPJOB (Display Job) 4-11
DSPJOBLOG (Display Job Log) 4-11
DSPOBJD (Display Object Description) 5-9
DSPPFM (Display Physical File Member) 4-20
DSPUSRPMN (Display User Permission) 6-24
EDTAUTL (Edit Authorization List) 6-7
EDTDLOAUT (Edit Document Library Object Authority) 6-25
EDTDOC (Edit Document) 8-26
ENDCALSRV (End Calendar Service) 2-12, D-74
ENDIXMON (End Index Monitor) 7-16, D-75

commands (continued)

FILDOC (File Document) 5-29
 FILLFORM (Fill Form Document) 8-27, D-76
 GRTUSRPMN (Grant User Permission) 6-26
 MOVDOC (Move Document) 5-33
 MOVOBJ (Move Object) 4-6
 MRGDOC (Merge Document) 8-29, D-79
 OVRDBF (Override Database File) 5-21
 OVRPRTF (Override Print File) A-3, A-4
 PAGDOC (Paginate Document) 8-32, D-84
 PRTCAL (Print Calendar) 2-12, D-85
 PRTDOC (Print Document) 8-33
 PRTSWL (Print Stop Word List) 7-13, 8-38
 QRYCALITM (Query Calendar Item) 2-13, D-94
 QRYDOCLIB (Query Document Library) 5-36
 QRYDST (Query Distribution) 4-20
 RCLDLO (Reclaim Document Library Object) 9-5
 RCV DST (Receive Distribution) 4-22
 RCVMSG (Receive Message) 10-3
 RGZDLO (Reorganize Document Library Object) 9-6
 RGZPFM (Reorganize Physical File Member) 9-9
 RMVCALITM (Remove Calendar Item) 2-15, D-104
 RMVDIRE (Remove Directory Entry) 3-15
 RMVDLOAUT (Remove Document Library Object Authority) 6-26
 RMVD OCCVN (Remove Document Conversion) D-107
 RMVDSTLE (Remove Distribution List Entry) 3-24
 RMVOFCAPPE (Remove Office Application Entry) 11-13, D-108
 RMVOFCENR (Remove Office Enrollment) 10-9, D-109
 RMVXTIDXE (Remove Text Index Entry) 7-17, D-110
 RNMDIRE (Rename Directory Entry) 3-16
 RNMDLO (Rename Document Library Object) 5-38
 RPLDOC (Replace Document) 5-39
 RSTCAL (Restore Calendar) 2-16, D-113
 RSTDLO (Restore Document Library Object) 5-14
 RTVDLONAM (Retrieve Document Library Object Name) 5-40
 RTVDOC (Retrieve Document) 5-42
 RTVOFCA (Retrieve Office Attributes) 10-3, D-117
 RTVOFCENR (Retrieve Office Enrollment) 10-10, D-119
 RTVSWLSRC (Retrieve Stop Word List Source) 7-13, 8-38
 RVKUSRPMN (Revoke User Permission) 6-27
 SAVCAL (Save Calendar) 2-17, D-122
 SAVDLO (Save Document Library Object) 5-14
 SAVLIB (Save Library) 1-4
 SNDDOC (Send Document) 4-25, D-127
 SNDDST (Send Distribution) 4-26
 STRCALSRV (Start Calendar Service) 2-18, D-128
 STRIDXMON (Start Index Monitor) 7-18, D-129

commands (continued)

STROFC (Start Office) 10-2, D-130
 STRPRSDIR (Start Personal Directory) D-131
 STRRGZIDX (Start Reorganization of Index) 7-18, D-132
 STRUPDIDX (Start Update of Index) 7-18, D-133
 STRWP (Start Word Processing) 8-38, D-134
 WRKDEVD (Work with Device Description) A-2
 WRKDIR (Work with Directory) 3-17
 WRKDOC (Work with Documents) 8-39
 WRKDOCCVN (Work with Document Conversions) 11-9, D-135
 WRKDOCLIB (Work with Document Library) 10-6
 WRKDOC PRTQ (Work with Document Print Queue) 8-41
 WRKDOCSET (Work with Document Sets) 11-13, D-136
 WRKDOCTYP (Work with Document Types) 11-13, D-137
 WRKDSTL (Work with Distribution List) 3-25
 WRKFLR (Work with Folders) 8-42
 WRKOF CAPPD (Work with Office Application Descriptions) 11-13, D-138
 WRKOF CAPPE (Work with Office Application Entries) 11-13, D-139
 WRKXTIDX (Work with Text Index) 7-19, D-140
 WRKXTPRF (Work with Text Profiles) 8-43, D-141

commitment control 9-10**compiling CL programs B-2****Configure VM/MVS Bridge (CFGRPDS)****command 10-1****confirmation mail log 4-2****considerations**

character set and code page A-1
 document list 5-10
 security for directory services 3-7
 service administration 1-3
 word processing character sets and code pages A-3

Control Calendar (QOCTLCL) API 2-28**Control Language (CL) commands**

character set considerations 2-4
 code page considerations 2-4
 creating source B-6
 creating the command B-7
 example command source B-7
 licensed program requirements 1-5
 parameters
 authority 2-4
 calendar group 2-3
 calendar name 2-3
 command character identifier 2-4
 date 2-3
 item identifier 2-3
 library 2-2
 time 2-3

Control Language (CL) programs

- compiling B-2
- creating B-1
- creating physical files B-3
- debugging B-3
- editing B-1
- example program B-4
- file considerations B-2
- running the program B-3
- statements B-1

Convert Document (CVTDOC) command D-55

Copy Document (CPYDOC) command 5-12, 5-17

Copy From PC Document (CPYFRMPCD) command 5-12, 5-20

Copy Spooled File (CPYSPLF) command 4-66

Copy To PC Document (CPYTOPCD) command 5-12, 5-21

Create Calendar (CRTCAL) command 2-8, D-44

Create Distribution List (CRTDSTL) command 3-20

Create Document (CRTDOC) command 8-16

Create Document Set (CRTDOCSET) command D-50

Create Document Type (CRTDOCTYP) command D-51

Create Folder (CRTFLR) command 5-12, 5-23

Create Office Application Description (CRTOFFCAPPD) command D-54

Create Physical File (CRTPF) command 5-20

Create Source Physical File (CRTSRCPF) command 5-20

Create Spelling Aid Dictionary (CRTSPADCT) command 7-14, 8-17

customer response 8-62

customized billing example 8-60

D

damaged database files 9-13

damaged objects 4-10

data management

- backup 9-1
- commitment control and journal management 9-10
- compressing database files 9-9
- data storage and journal management 9-8
- individual object restore 9-3
- managing storage used by database objects and journals 9-8
- managing storage used by distribution objects 9-11
- managing storage used by documents and folders 9-8
- OfficeVision/400 object individual recovery
 - damaged database files 9-13
 - damaged QUSRSYS or QDOC system library 9-15
 - document 9-11
 - folders 9-12
 - internal document library objects 9-12

data management *(continued)*

- OfficeVision/400 object individual recovery *(continued)*
 - logically damaged documents or folders 9-12
 - QDOC and QUSRSYS library mismatches 9-15
- restoring
 - database files 9-4
 - individual documents and folders 9-3

database file

- compressing 9-9
- creating 8-4
- damaged 9-13

Delete Calendar (DLTCAL) command 2-10, D-56

Delete Distribution (DLTDST) command 4-19

Delete Distribution List (DLTDSTL) command 3-22

Delete Document Library Object (DLTDLO) command 5-13, 5-24

Delete Document Set (DLTDOCSET) command D-58

Delete Document Type (DLTDOCTYP) command D-59

Delete Journal Receiver (DLTJRNRCV) command 9-11

Delete Office Application Description (DLTOFCAPPD) command D-60

Delete Outgoing Mail (DLTOUTMAIL) command 10-3, D-61

Delete Spelling Aid Dictionary (DLTSPADCT) command 8-20

designing help 8-9

destination system

- description 4-1

DIA (Document Interchange Architecture) dictionary 4-1

- description 8-2
- language 8-3
- naming conventions 8-6
- spelling aid dictionary
 - creating 8-17
 - deleting 8-20
- user 8-4

directory files 3-3

directory services

- command interfaces 3-5
- commands 1-6
- control language interfaces 3-10
- directory files 3-3
- disk unit error 4-10
- example applications 3-26
- object naming conventions 3-5
- objects 3-3
- other topics 3-7
- output file 3-5
- overview 3-1
- security considerations 3-7
- updating the directory from customer data 3-32

directory services *(continued)*

user restrictions 3-8

Display Authorization List Document Library

Objects (DSPAUTLDLO) command 6-22

Display Calendar Authority (DSPCALAUT) command

calendar services 2-11

description D-62

output file considerations C-2

Display Calendar Description (DSPCALD) command

calendar services 2-11

description D-65

output file considerations C-4

Display Calendar Item (DSPCALITM) command

calendar services 2-12

description D-68

output file considerations C-7

Display Directory (DSPDIR) command

directory services 3-13

output file considerations C-11

Display Distribution List (DSPDSTL) command

directory services 3-23

output file considerations C-40

Display Document (DSPDOC) command 8-22

Display Document Conversion (DSPDOCCVN) command

description D-71

output file considerations C-42

Display Document Library Object Auditing level

(DSPDLOAUD) command 6-23

output file considerations C-39

Display Document Library Object Authority

(DSPDLOAUT) command 6-23

Display Document Library Object Name

(DSPDLONAM) command 5-13, 5-26

Display File Description (DSPFD) command 9-9

Display Folder (DSPFLR) command

document library services 5-13, 5-27

output file considerations

document object information C-45

folder object information C-43

Display Help Document (DSPHLPDOC)

command 8-23

Display Index Status (DSPIDXSTS) command 7-16, D-73

Display Job (DSPJOB) command 4-11

Display Job Log (DSPJOBLOG) command 4-11

Display Object Description (DSPOBJD)

command 5-9

Display Physical File Member (DSPPFM)

command 4-20

Display User Permission (DSPUSRPMN)

command 6-24

distribution

document unit object 4-5

list commands 3-19

list files 3-5

distribution *(continued)*

lists

adding to 3-19

creating 3-20

deleting 3-22

displaying 3-23

explanation of 3-2

maintaining multiple 3-26

removing entries 3-24

work with distribution list command 3-25

maintaining multiple lists 3-26

services 4-1

tracking object 4-6

distribution document

changing 4-18

deleting 4-19

querying 4-20

receiving 4-22

sending 4-25

sending messages 4-26

distribution list

definition 3-1

distribution recipient index (DRX) 3-5, 4-6

distribution tracking object (DTO) 4-6

document

attributes 5-5

changing 5-12

checking 5-12

content 5-5

conversions 11-9

copy to a personal computer 5-12

copying 5-12

copying from a personal computer 5-12

creating 8-16

deleting 5-24

description 6-12

description records C-51

details 5-5

editing 8-26

filing 5-29

fillform 8-27

interchange 8-8

list considerations 5-10

merging 8-29

moving 5-33

paginating 8-32

password 6-10

printing 8-33

querying 5-36

renaming 5-38

retrieving 5-42

spell checking 8-14

unit object 4-6

Document Commands (CMDDOC) menu 5-4

document distribution services

application interfaces 4-9

document distribution services *(continued)*

- commands 1-9
- control language interfaces 4-17
- damaged objects 4-10
- distribution recipient index 4-6
- example applications 4-29
- explanation of 4-1
- jobs in the QSNADS subsystem 4-10
- mail logs 4-2
- managing mail 4-1
- objects 4-5
- OSI message service/400 4-3
- other topics 4-10
- overview 4-1
- receiving mail 4-1
- relationship to SNADS 4-3
- restrictions 4-11
- security considerations 4-17
- sending mail 4-1
- simple mail transfer protocol (SMTP) 4-4
- system messages 4-12
- VM/MVS bridge 4-3
- working on behalf of another user 4-17

document handling exit 11-9**Document Interchange Architecture (DIA)** 4-1**document library**

- definition 5-2

Document Library Object (CMDDLO) menu 5-4**document library objects**

- adding 6-19
- changing ownership 6-20
- definition 5-5
- display authority list 6-22

document library services

- access codes 5-10
- accessing 5-2
- applications E-1
- character set and code page considerations 5-11
- commands 1-9
- control language interfaces 5-12
- definition 5-1
- document content 5-5
- document details 5-5
- document lists 5-6
- documents 5-5
- enrollment requirements 5-9
- example applications 5-45
- folder security on personal computer 5-10
- folders lists 5-6
- hierarchical file system E-1
- host display station 5-3
- lists 5-6
- locally attached PC interface 5-4
- object naming 5-8
- object ownership 5-8
- objects 5-5

document library services *(continued)*

- OS/400 CL commands 5-5
- overview 5-1
- PC files and folders 5-7
- remotely attached system interface 5-4
- security considerations 5-9
- working on behalf of another user 5-10

document list

- considerations 5-10
- definition 5-6

document unit object

- definition 4-6

DRX (distribution recipient index) 3-5, 4-6**DTO (distribution tracking object)** 4-6**Dump Document Library Object (DMPDLO)****command** 5-13**Dump System Object (DMPSYSOBJ) command** 4-7**E****Edit Authorization List (EDTAUTL) command** 6-7**Edit Document (EDTDOC) command** 8-26**Edit Document Library Object Authority****(EDTDLOAUT) command** 6-25**editor selection** 8-7**electronic mail**

- sending 4-1
- simple mail transfer protocol (SMTP) 4-4

End Calendar Service (ENDCALSRV)**command** 2-12, D-74**End Index Monitor (ENDIDXMON) command** 7-16, D-75**enhancing OfficeVision/400**

- AS/400 business graphics utility application programming interfaces 11-5
- AS/400 query 11-5
- communications 11-8
- database files 11-7
- graphical data display manager commands 11-6
- high-level language support 11-8
- OfficeVision/400 application enabler 11-8, 11-12
- operating system group job support (suspend/resume) 11-1
- PC Support/400 11-7

enrollment

- commands 10-7
- considerations 1-3
- definition 1-3
- requirements 5-9

error mail log 4-2**example applications**

- building a bridge between office and object distribution 4-88
- calendar user-defined function 2-24
- Control Calendar (QOCTLCL) API 2-30
- converting a spooled file to FFTDCA 4-61

example applications *(continued)*

- creating customized billings 8-60
 - Creating fill form document 8-45
 - customer response 8-62
 - delegating mail 4-40
 - deleting outgoing mail 4-96
 - displaying calendar items matching a query 2-25
 - document distribution services 4-29
 - document library services 5-45
 - ending mail delegate 4-55
 - granting user permission 4-29
 - identifying new mail 4-95
 - initial program to call STROFC or another user menu 10-13
 - maintaining mail logs 4-73
 - maintaining multiple distribution lists 3-26
 - printing all documents in a folder 8-54
 - printing repetitive letters 8-60
 - programs to query a user's mailbox 4-60
 - receiving messages, notes and documents 4-36
 - saving final form documents 8-57
 - sending messages, notes and documents to OfficeVision/400 users 4-33
 - spell checking changed documents 8-56
 - STROFC command 10-10
 - updating the directory from customer data 3-26
 - user permission 6-10
 - user profile maintenance 3-26
 - using CHKDLO to determine authority 6-28
- *EXCLUDE authority 6-13**

F

File Document (FILDOC) command

- document library services 5-13, 5-29
- output file considerations C-47

file format for PRTFILE files (Office)

- document description records C-51
- general format record C-55
- page format record C-56

Fill Form Document (FILLFORM) command 8-27, D-76

- output file considerations description C-47

Folder Commands (CMDFLR) menu 5-4

folders

- checking 5-15
- creating 5-23
- definition 5-7
- deleting 5-24
- displaying 5-27
- renaming 5-38

folders and document lists 5-6

four levels of authority 6-13

G

GDDM (graphical data display manager) 11-6

general format record C-55

grade level analysis on documents 8-14

Grant User Permission (GRTUSRPMN)

command 6-26

graphical data display manager (GDDM) 11-6

group

- authorization 6-3

- profiles 6-3

GRPCALS parameter 2-3

H

help

- accessing 8-12

- text documents 8-5, 8-9

HFS Control File System (QHCTLFS) API E-7

hierarchical file system (HFS) E-1

high-level language support 11-8

host display station interface 5-3

I

in-mail log 4-2

indexing

- text search 7-1, 7-3

indirect user 3-1

interchanging documents 8-8

internal mail logs

- dumping 4-7

J

journal management 9-8

L

LADN (library-assigned document name) 5-9

language dictionaries 7-2, 7-7, 8-3

language identifier 7-2, 7-12

levels of authority 6-13

library-assigned document name (LADN) 5-9

licensed program requirements 1-5

line format record C-57

linguistic processing 7-2, 7-8

list ID

- definition 3-8

local user 3-1

locally attached PC interface 5-4

logically damaged documents or folders 9-12

M

mail

- delegating 4-40

mail (continued)

- deleting outgoing 4-96
- dumping internal mail logs 4-7
- ending delegate 4-55
- logs 4-2, 4-7
- maintaining mail logs 4-73
- new indicator example 4-95
- sending 4-1
- simple mail transfer protocol (SMTP) 4-4
- using document distribution services 4-1

mail log

- definition 4-2

main menu

- customizing 1-16

maintaining multiple distribution lists example 3-26

menus

- customizing 1-16
- Document Commands (CMDDOC) 5-4
- Document Library Object (CMDDLLO) 5-4
- Folder Commands (CMDFLR) 5-4
- Office Commands (CMDOFC) 5-4

Merge Document (MRGDOC) command 8-29, D-79

merging and printing text and data 8-1

messages

- receiving 4-36
- sending 4-26, 4-33

miscellaneous services

- commands 1-14
- overview 10-1

Move Document (MOVDOC) command 5-13, 5-33

Move Object (MOV OBJ) command 4-6

multiple distributions list, maintaining 3-26

N

naming conventions 3-5, 8-6

national language dictionaries 7-2

notepads 8-5

notes

- receiving 4-36
- sending 4-33

O

object

- authority 6-1
- damaged 4-10
- distribution 3-7
- naming conventions 3-5, 8-6
- ownership 5-8
- recovery 9-11

object naming conventions

- directory services 3-5
- word processing services 8-6

Office Commands (CMDOFC) menu 5-4

office services

- base 1-1
- command interfaces
 - calendar services 2-1
 - description 1-4
 - directory services 3-1
 - document distribution services 4-1
 - document library services 5-1
 - security services 6-1
 - word processing services 8-1
- description 1-1
- optional 1-1
- output file considerations
 - Add Calendar Item (ADDCALITM) command C-1
 - Display Calendar Authority (DSPCALAUT) command C-2
 - Display Calendar Description (DSPCALD) command C-4
 - Display Calendar Item (DSPCALITM) command C-7
 - Display Directory (DSPDIR) command C-11
 - Display Distribution List (DSPDSTL) command C-40
 - Display Document Conversion (DSPDOCCVN) command C-42
 - Display Folder (DSPFLR) command C-43
 - File Document (FILDOC) command C-47
 - Print Document (PRTDOC) command C-49
 - Query Calendar Item (QRYCALITM) command C-64
 - Query Distribution (QRYDST) command C-76
 - Query Document Library (QRYDOCLIB) command C-69
 - Receive Distribution (RCVDST) command C-87
 - Replace Document (RPLDOC) command C-93
 - Retrieve Document (RTVDOC) command C-94
 - Save Document Library Object (SAVDLO) command C-100
 - Send Distribution (SNDDST) command C-104
- relationship to other products 1-2
- using with user applications 1-18

OfficeVision/400

- command interfaces 1-1
- customizing the main menu 1-16
- enhancing 11-1
- integrating applications using commands 1-16
- office services 1-4
- overview 1-1
- relationships between services and products 1-2
- starting 5-3
- suspend/resume function 11-1
- using functions with user application menus 1-16
- using services with user applications 1-18

OfficeVision/400 application enabler

- ADDDOCCVN (Add Document Conversion) command 11-9

OfficeVision/400 application enabler *(continued)*

- definition 11-8, 11-12
- document handling exit 11-9
- PCFI0000 11-19
- performance considerations 11-12, 11-16
- processing considerations 11-11, 11-15
- QO1PCFID (PCFILE Identification Process)
 - API 11-17
- QOGCHGOE (Change Office Exit Programs)
 - API 11-10
- QOGRTVOE (Retrieve Office Exit Programs)
 - API 11-11, 11-15
- WRKDOCCVN (Work with Document Conversions)
 - command 11-9

OfficeVision/VM 1-3, 4-3**online help**

- designing 8-9

operating system group job support 11-1**operational setup**

- APPC devices 1-4
- QSNADS subsystem 1-4
- SNADS network 1-4

OSI message service/400 4-3**OUTFILFMT parameter 3-4****output files**

- considerations for OfficeVision/400 commands
 - Add Calendar Item (ADDCALITM) command C-1
 - Display Calendar Authority (DSPCALAUT)
 - command C-2
 - Display Calendar Description (DSPCALD)
 - command C-4
 - Display Calendar Item (DSPCALITM)
 - command C-7
 - Display Directory (DSPDIR) command C-11
 - Display Distribution List (DSPDSTL)
 - command C-40
 - Display Document Conversion (DSPDOCCVN)
 - command C-42
 - Display Document Library Object Auditing Level (DSPDLOAUD) command C-39
 - Display Folder (DSPFLR) command C-43
 - File Document (FILDOC) command C-47
 - Fill Form Document (FILLFORM)
 - command C-47
 - Print Document (PRTDOC) command C-49
 - Query Calendar Item (QRYCALITM)
 - command C-64
 - Query Distribution (QRYDST) command C-76
 - Query Document Library (QRYDOCLIB)
 - command C-69
 - Receive Distribution (RCVDST) command C-87
 - Replace Document (RPLDOC) command C-93
 - Retrieve Document (RTVDOC) command C-94
 - Save Document Library Object (SAVDLO)
 - command C-100
 - Send Distribution (SNDDST) command C-104

output files *(continued)*

- naming conventions
 - calendar services 2-2
 - using with directory services 3-4
- Override Database File (OVRDBF) command 5-21**
- Override Print File (OVRPRTF) command A-3, A-4**
- owner**
 - definition 5-8
- ownership rights 6-4**

P

- page format record C-56**
- Paginate Document (PAGDOC) command 8-32, D-84**
- password 6-10**
- PC files and folders 5-7**
- PC interface**
 - locally attached 5-4
- PC Support/400**
 - description 11-7
- PCFI0000 (PCFILE User Program Interface) 11-19**
- PCFILE Identification Process (QO1PCFID)**
 - API 11-17
- PCFILE User Program Interface (PCFI0000) 11-19**

personal

- document 6-26
- folder 6-26
- items 6-9
- mail 6-26

physical files

- creating B-3

polling frequency 7-5**precedence rules 6-8****primary index 7-3****Print Calendar (PRTCAL) command 2-12, D-85****Print Document (PRTDOC) command**

- output file considerations
 - description C-49
 - PRTFILE file formats C-50
 - text lines C-58
- word processing services 8-33

Print Stop Word List (PRTSWL) command 7-13, 8-38**printing**

- documents in a folder 8-54
- repetitive letters 8-60
- text and data 8-1

profiles 6-3**PRTFILE file formats C-50****public authorization 6-4****Q****QDFTOWN entry 3-9**

QDIA job 4-10
QDIAINDUSR job 4-11
QDIALOCAL job 4-10
QDIANOTIFY job 4-11
QDOC entry 3-9
QHFCTLFS (HFS Control File System) API E-7
QLPAUTO entry 3-9
QLPINSTL entry 3-9
QO1PCFID (PCFILE Identification Process) API 11-17
QOGCHGOE (Change Office Exit Programs) API 11-10
QOGRTVOE (Retrieve Office Exit Programs) API 11-11, 11-15
QOCTLCL (Control Calendar) API 2-28
QSECOFR entry 3-9
QSYS entry 3-9
queries 8-5
Query Calendar Item (QRYCALITM) command
 calendar services 2-13
 description D-94
 output file considerations C-64
Query Distribution (QRYDST) command
 document distribution services 4-20
 output file considerations
 description C-76
 incoming distribution information C-76
 outgoing distribution information C-82
Query Document Library (QRYDOCLIB) command
 document library services 5-13, 5-36
 output file considerations C-69
query user's mailbox 4-60
QUSER entry 3-9
QUSRSYS library 1-4, 3-5

R
Receive Distribution (RCVDST) command
 document distribution services 4-22
 output file considerations C-87
Receive Message (RCVMSG) command 10-3
recipient 4-1
Reclaim Document Library Object (RCLDLO) command
 data management 9-5
 document library services 5-13
recovery of data 9-1, 9-11
related printed information F-1
remote document library processing 10-4
remote user 3-1
remotely attached system interface 5-4
remotely filed documents 6-15
Remove Calendar Item (RMVCALITM) command 2-15, D-104
Remove Directory Entry (RMVDIRE) command 3-15

Remove Distribution List Entry (RMVDSTLE) command 3-24
Remove Document Conversion (RMVDOCCVN) command D-107
Remove Document Library Object Authority (RMVDLOAUT) command 6-26
Remove Office Application Entry (RMVOFCAPPE) command D-108
Remove Office Enrollment (RMVOFCENR) command 10-9, D-109
Remove Text Index Entry (RMVTXTIDX) command 7-17, D-110
Rename Directory Entry (RNMDIRE) command 3-16
Rename Document Library Object (RNMDLO) command 5-13, 5-38
Reorganize Document Library Object (RGZDLO) command
 data management 9-6
 document library services 5-13
Reorganize Physical File Member (RGZPFM) command 9-9
reorganize text index 7-2
Replace Document (RPLDOC) command
 document library services 5-13, 5-39
 output file considerations C-93
Restore Calendar (RSTCAL) command 2-16, D-113
Restore Document Library Object (RSTDLO) command 5-14
restoring
 individual documents and folders 9-3
 individual objects 9-3
Retrieve Document (RTVDOC) command
 document library services 5-13, 5-42
 output file considerations C-94
Retrieve Document Library Object Name (RTVDLONAM) command 5-13, 5-40
Retrieve Office Attributes (RTVOFCA) command 10-3, D-117
Retrieve Office Enrollment (RTVOFCENR) command 10-10, D-119
Retrieve Office Exit Programs (QOGRTVOE) API 11-11, 11-15
Retrieve Stop Word List Source (RTVSWLSRC) command 7-13, 8-38
Revoke User Permission (RVKUSRPMN) command 6-27
RSCS network 4-3
run instruction 8-13

S
save
 final form documents 8-57
 system authority 6-1, 6-6
Save Calendar (SAVCAL) command 2-17, D-122

Save Document Library Object (SAVDLO) command

document library services 5-14

output file considerations

description C-100

folder object information C-100

Save Library (SAVLIB) command 1-4**Scanmaster 4-12****scheduling calendar**

definition 2-1

scheduling queue 7-1, 7-4**security administrator special authority 3-9, 6-1, 6-5****security considerations**

directory services 3-7

document distribution services 4-17

document library 5-9

security services

access codes 6-18

access control functions 6-2

AS/400 enrollment 6-3

audit objects 6-12

authorization lists 6-7

CL interfaces 6-19

commands 1-10

document and folder 6-12

document library objects 6-12

group authorization 6-3

group profiles 6-3

overview 6-1

ownership rights 6-4

public authorization 6-4

special authority 6-5

specific user authorization 6-3

user profiles 6-3

working on behalf of another user 6-9

Send Distribution (SNDDST) command

document distribution services 4-26

output file considerations C-104

Send Document (SNDDOC) command 4-25, D-127**sender 4-1****service administration considerations**

enrollment 1-3

operational setup 1-4

simple mail transfer protocol (SMTP) 4-4**SMTP 4-4****SNA distribution services (SNADS) 1-2, 4-1****SNADS (SNA distribution services) 1-2, 4-1****source**

creating for CL commands B-6

creating for programs B-1

SPCAUT parameter 3-9**special authority 6-1, 6-5****specific user authorization 6-3****spell checking changed documents 8-56****spelling aid dictionary**

creating 8-17

spelling aid dictionary (continued)

deleting 8-20

spooled file

converting to FFTDCA 4-61

Start Calendar Service (STRCALSRV)

command 2-18, D-128

Start Index Monitor (STRIDXMON) command 7-18, D-129**Start Office (STROFC) command**

description D-130

examples 10-10

miscellaneous services 10-2

Start Personal Directory (STRPRSDIR) command

description D-131

Start Reorganization of Index (STRRGZIDX)

command 7-18, D-132

Start Update of Index (STRUPDIDX)

command 7-18, D-133

Start Word Processing (STRWP) command 8-38, D-134**stop word list**

creating 7-14

CRTSPADCT (Create Spelling Aid Dictionary)

command 7-14, 8-17

description 7-13

PRTSWL (Print Stop Word List) command 7-13

RTVSWLSRC (Retrieve Stop Word List Source)

command 7-13

user dictionaries 8-4

storage management

used by database objects and journals 9-8

used by distribution objects 9-11

used by documents and folders 9-8

storage management of data 9-1**substitution variables 2-20****suspend/resume function 11-1****system authority levels 6-1****system distribution directory**

adding new entries to 3-10

changing attributes 3-12

changing entries 3-12

definition 1-3

description 3-2

displaying entries 3-13

entries 3-9

removing entries 3-24

using the work with directory command 3-25

T**text index 7-1, 7-3****text lines C-58****text recovery 8-8****text search services**

additional index 7-3

administration 7-2

text search services *(continued)*

- administration monitor 7-5
- backup and recovery 7-7
- character set and code page 7-16
- commands 1-11, 7-15
- files 7-5
- indexing 7-1, 7-3
- language dictionaries 7-7
- linguistic processing 7-2, 7-8
- objects 7-3
- overview 7-1
- primary index 7-3
- scheduling queue 7-1, 7-4
- search function 7-3
- stop word list 7-13
- using dictionaries 7-2

U

update text index 7-2

***USE authority** 6-13

user

- application menus 1-16
- applications 1-18
- assigned name 5-9
- authorizing 6-1
- document library 5-2
- ID 3-2
- permission example 6-10
- profile maintenance example 3-26
- profiles 6-3
- restrictions 3-8
- types 3-1

user permission

- commands 6-10
- example of granting 4-29
- revoking 6-27

user profile 6-3

V

VM/370 RSCS network 4-3

VM/MVS bridge

- configuring 10-1
- definition 11-8
- description 4-3
- support 1-3
- using to send distributions 4-1

W

word processing document functions 8-1

word processing services

- accessing help text 8-12
- application program interfaces 8-6
- character set and code page considerations A-3

word processing services *(continued)*

- commands 1-13
- control language commands 8-6
- control language interfaces 8-14
- database files 8-4
- defining help areas for application display file 8-10
- designing online help with help text documents 8-9
- dictionaries 8-2, 8-3
- document functions 8-1
- editor selection 8-7
- example applications 8-45
- help text documents 8-5
- interchanging documents 8-8
- notepads 8-5
- object naming conventions 8-6
- other topics 8-7
- overview 8-1
- printing and merging text and data 8-1
- queries 8-5
- run instruction 8-13
- security considerations 8-14
- service objects 8-2
- spell checking changed documents 8-56
- starting 8-38
- text profiles 8-2
- text recovery 8-8
- user dictionaries 8-4
- viewing a help text document 8-12

work on behalf of another user 4-17, 6-9

Work with Device Description (WRKDEVD)
command A-2

Work with Directory (WRKDIR) command 3-17

Work with Distribution List (WRKDSTL)
command 3-25

Work with Document Conversions (WRKDOCCVN)
command 11-9, D-135

Work with Document Library (WRKDOCLIB)
command 10-6

Work with Document Print Queue (WRKDOCPRTQ)
command 8-41

Work with Document Sets (WRKDOCSET)
command D-136

Work with Document Types (WRKDOCTYP)
command D-137

Work with Documents (WRKDOC) command 8-39

Work with Folders (WRKFLR) command 8-42

Work with Office Application Descriptions (WRKOFCAAPPD) command D-138

Work with Office Application Entries (WRKOFCAPPE) command D-139

Work with Text Index (WRKTXTIDX)
command 7-19, D-140

Work with Text Profiles (WRKTXTPRF)
command 8-43, D-141

X

X.400 4-3

Reader's Comments

Application System/400
Office Services Concepts and
Programmer's Guide
Version 2

Publication No. SC41-9758-02

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 | | | | |
| THANK YOU! | | | | |

Please tell us how we can improve this manual:

May we contact you to discuss your responses? Yes No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it: **(+1)+817+962-3470**
- 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. _____



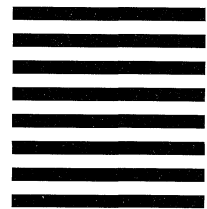
Fold and Tape

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 5D9
IBM CORPORATION
5 WEST KIRKWOOD BOULEVARD
ROANOKE TX 76299-0001



Fold and Tape

Please do not staple

Fold and Tape

Reader's Comments

Application System/400
Office Services Concepts and
Programmer's Guide
Version 2

Publication No. SC41-9758-02

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 | | | | |
| THANK YOU! | | | | |

Please tell us how we can improve this manual:

May we contact you to discuss your responses? Yes No

Phone: (____) _____ Fax: (____) _____

To return this form:

- Mail it
- Fax it: **(+1)+817+962-3470**
- 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 5D9
IBM CORPORATION
5 WEST KIRKWOOD BOULEVARD
ROANOKE TX 76299-0001



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5738-SS1

Printed in Denmark by Bonde's

SC41-9758-02

