

DITTO/ESA



User's Guide and Reference

Release 3

DITTO/ESA



User's Guide and Reference

Release 3

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 373.

Third Edition (June 1999)

This edition applies to Version 1 Release 3 Modification Level 0 of IBM Data Interfile Transfer, Testing, and Operations Utility for ESA, program numbers 5655-103 (OS/390® or MVS™), 5654-029 (VM), 5648-099 (VSE), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

This publication is also available in various formats on the Web at:

<http://www.software.ibm.com/ad/ditto/>

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, Department HHX/H3
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

or use the form on the Web at:

<http://www.software.ibm.com/ad/rcf/>

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

© Copyright International Business Machines Corporation 1995, 1999. All rights reserved.

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

Contents

| | |
|---|------|
| About This Book | xiii |
| Other Sources of Information on DITTO | xiii |
| Summary of Changes | xv |
| Major Changes to DITTO/ESA | xv |
| Release 3, June 1999 | xv |
| Release 2, December 1996 | xvi |
| Release 1, April 1995 | xvii |
| Changes to This Book | xx |
| Third Edition (June 1999) | xx |

Part 1. Introduction to DITTO 1

| | |
|---|----|
| Chapter 1. Overview of DITTO Functions | 3 |
| Where You Can Run DITTO | 3 |
| What Types of Data You Can Work With | 3 |
| What You Can Do With Data | 3 |
| Formats for Displaying or Printing Data | 4 |
| Looking at Data | 5 |
| Changing Data | 6 |
| Copying Data | 7 |
| Printing Data | 8 |
| Erasing Data | 8 |
| Finding Data | 8 |
| Working with a List of Data Sets, Members, or Objects | 9 |
| Working with Tapes | 9 |
| Querying and Changing Your Processing Environment | 10 |
| Chapter 2. Running DITTO in Different Modes | 11 |
| Running DITTO in Full-Screen Mode | 11 |
| MVS or CMS: Starting DITTO in Full-Screen Mode | 11 |
| VSE: Starting DITTO in Full-Screen Mode | 12 |
| CMS/VSE Mode: Starting VSE/DITTO from CMS | 14 |
| The DITTO Full-Screen Panels | 15 |
| Selecting a Function | 17 |
| MVS: Running DITTO under ISPF | 18 |
| CMS: Running CMS in Subset Mode from DITTO | 18 |
| Switching between Full-Screen Mode and Line Mode | 18 |
| Handling Errors | 19 |
| Example of a Full-Screen Session | 19 |
| Running DITTO in Line Mode | 19 |
| Starting DITTO in Line Mode | 20 |
| Entering Commands in Line Mode | 20 |
| VSE: Running DITTO in a VSE/ESA Dynamic Partition | 21 |
| Handling Errors | 21 |
| Example of a Line Mode Session | 21 |
| Executing a Single DITTO Function in Command Mode | 21 |
| Deciding Which Syntax to Use | 23 |
| Example of a Command Mode Procedure | 23 |

| | |
|---|----|
| Running DITTO in Batch Mode | 24 |
| MVS: Running DITTO in Batch Mode | 24 |
| VSE: Running DITTO in Batch Mode | 24 |
| CMS: Running DITTO in Batch Mode | 25 |
| Coding DITTO Control Statements | 25 |
| Handling Errors | 25 |
| Examples of a Batch Job | 25 |
| DITTO Print Output | 26 |
| VSE: System File Assignments | 27 |
| VSE: Functions Supported in Stand-Alone Environment | 27 |

Part 2. Guide to Using DITTO Functions 29

| | |
|--|-----------|
| Chapter 3. Browsing Data in Full-Screen Mode | 31 |
| Browsing Data | 32 |
| Browsing a Collection of Records | 33 |
| Changing the Data Display Format | 33 |
| Locating a Specific Record | 33 |
| Seeing More of a Record | 35 |
| Saving Data in a Buffer | 35 |
| Printing Data | 36 |
| Purging Print Output and Spool Files | 36 |
| Leaving the Browse Function | 36 |
| Browsing a Collection of Bytes | 36 |
| Changing the Data Display Format | 37 |
| Locating Specific Data | 37 |
| Saving Data in a Buffer | 38 |
| Printing Data | 38 |
| Leaving the Browse Function | 38 |
| Chapter 4. Editing Data in Full-Screen Mode | 39 |
| Editing Data | 40 |
| Changing the Data Display Format | 41 |
| Locating a Specific Record | 41 |
| Seeing More of a Record | 42 |
| Changing the Length of a Record (DTE and VE Only) | 42 |
| Setting the Length for Inserted Records (DTE and VE Only) | 43 |
| Adding a Key Field to a Disk Record (DTE Only) | 43 |
| Mapping the Track Contents (DTE Only) | 43 |
| Locating and Changing a String | 43 |
| Translating Input to Uppercase | 46 |
| Setting the Prefix Area Position | 46 |
| Recovering Previously Deleted Data | 47 |
| Clearing Commands in Prefix Area | 47 |
| Splitting and Joining Lines (LE and VE Only) | 47 |
| Sending Library Member Contents to POWER (LE Only) | 47 |
| Including the Contents of Another Library Member (LE Only) | 47 |
| Editing another Library Member without Exiting (LE Only) | 47 |
| Additional Commands for Single-Record Editing | 48 |
| Prefix Area Commands | 48 |
| Saving and Exiting | 49 |
| VSAM Edit Restrictions and Considerations | 49 |

| | |
|--|----|
| Chapter 5. Processing Catalog Entries | 51 |
| Printing a List of Catalog Entries | 52 |
| Working with Catalog Entries in Full-Screen Mode | 52 |
| Working with a List of Catalog Entries | 54 |
| Parameters for a Catalog Entry | 58 |
| Statistics for VSAM Data Sets | 60 |
| | |
| Chapter 6. Processing Tapes | 61 |
| Specifying Tape Input or Output | 62 |
| MVS: Specifying Tape Input or Output with JCL | 63 |
| VSE: Specifying Tape Input or Output with JCL | 64 |
| MVS: Specifying Tape Input or Output without JCL | 65 |
| VSE: Specifying Tape Input or Output without JCL | 66 |
| CMS: Specifying Tape Input or Output | 66 |
| Tape Density and Mode Values | 67 |
| VSE: Permanent or Temporary Tape Assignment | 68 |
| Tape Block Size Considerations | 68 |
| Positioning a Tape | 68 |
| How Tapes are Organized | 68 |
| Specifying the Number of Tape Files to Process | 70 |
| Processing Tapes with Data-Check Errors | 71 |
| Finding Out What a Tape Contains | 71 |
| Browsing and Updating Tape Files | 73 |
| Updating Tape Blocks in Full-Screen Mode | 74 |
| Updating One Physical Record at a Time | 75 |
| Viewing a Physical Tape Record at the System Console | 77 |
| Searching for Specific Data | 78 |
| Printing Tape Files | 79 |
| Printing SYSLST Tape Files (VSE Only) | 79 |
| Controlling Print Output | 79 |
| Copying Tape Records | 80 |
| Copying from Tape to Tape | 80 |
| Copying Tape Data across a Network | 81 |
| ASCII Translation | 81 |
| Comparing Tapes | 82 |
| Initializing a Tape | 82 |
| Creating Tape Records | 83 |
| Erasing Tape Records | 83 |
| Examples of DITTO Tape Functions | 84 |
| | |
| Chapter 7. Processing Physical Disk Records | 87 |
| Processing Disk Records | 88 |
| Specifying Disk Input and Output | 88 |
| MVS: Specifying Disk Extents | 88 |
| MVS: Using Disk Fullpack Processing | 89 |
| Listing the Files on a Disk | 89 |
| Using the DVT Function in Full-Screen Mode | 90 |
| VSE: Displaying the Contents of the Label Area | 92 |
| Locating a Disk Data Set | 92 |
| Browsing and Updating Disk Records | 93 |
| Updating Disk Records in Full-Screen Mode | 93 |
| Updating One Record at a Time | 93 |
| Viewing a Disk Record at the System Console | 94 |
| Printing Disk Records | 95 |

| | |
|---|------------|
| Specifying What You Want to Print | 95 |
| Controlling Print Output | 96 |
| Writing a Disk End-of-File Record | 96 |
| Displaying or Changing a Disk Volume ID under VSE or CMS | 97 |
| Searching for Specific Data | 97 |
| Command Mode Example Using DITTO Disk Functions | 98 |
| Batch Examples Using DITTO Disk Functions | 99 |
| | |
| Chapter 8. Processing Sequential Data (SAM and QSAM) | 101 |
| Specifying Sequential Input and Output | 102 |
| MVS: Specifying Sequential Input and Output with JCL | 102 |
| VSE: Specifying Sequential Input and Output with JCL | 102 |
| MVS: Specifying Sequential Input and Output without JCL | 103 |
| VSE: Specifying Sequential Input and Output without JCL | 104 |
| CMS: Specifying Sequential Input | 104 |
| Record Format and Block Size | 105 |
| Specifying Records to be Processed | 105 |
| Printing Sequential Data | 105 |
| MVS: Printing Physical Records | 105 |
| VSE: Printing SYSLST Disk Files | 105 |
| Controlling Print Output | 106 |
| Copying Sequential Data Records | 106 |
| MVS or VSE: Creating Records | 106 |
| Batch Examples Using DITTO Sequential Data Functions | 107 |
| | |
| Chapter 9. Processing VSAM Data Sets | 109 |
| Specifying VSAM Input and Output | 110 |
| Specifying VSAM Input and Output with JCL | 110 |
| Specifying VSAM Input and Output without JCL | 110 |
| CMS: Specifying the VSAM Master Catalog | 111 |
| Specifying Records to be Processed | 111 |
| Control Interval Access | 111 |
| Browsing and Updating VSAM Data Sets | 112 |
| Updating VSAM Data Sets in Full-Screen Mode | 112 |
| Updating One Record at a Time | 113 |
| Printing VSAM Data Sets | 113 |
| Controlling Print Output | 113 |
| Copying VSAM Records | 114 |
| Creating VSAM Records | 114 |
| Examples of DITTO VSAM Functions | 114 |
| | |
| Chapter 10. CMS: Processing CMS Files | 117 |
| Specifying CMS Input and Output | 118 |
| Browsing CMS Files | 118 |
| Printing CMS Files | 118 |
| Controlling Print Output | 118 |
| Copying CMS Files | 119 |
| Copying Files in CMS/VSE Mode | 119 |
| Dumping and Restoring CMS Files | 119 |
| Dumping Files to Tape | 119 |
| Loading Files from a DITTO Dump Tape | 120 |
| Loading Files from a DDR or Fullpack DDR Tape or File | 120 |
| | |
| Chapter 11. VSE: Processing Library Members | 123 |

| | |
|---|------------|
| Specifying Library Input and Output | 124 |
| Listing the Members in a Library | 124 |
| Using the LDL Function in Full-Screen Mode | 125 |
| Listing Status Information for a Library | 127 |
| Using the LSL Function in Full-Screen Mode | 127 |
| Browsing and Updating Library Members in Full-Screen Mode | 129 |
| Printing Library Members | 130 |
| Controlling Print Output | 130 |
| Copying Library Members | 130 |
| Deleting or Renaming Library Members | 131 |
| Examples of DITTO Library Functions | 131 |
| | |
| Chapter 12. MVS: Processing OAM Objects | 133 |
| An Introduction to the Object Access Method | 134 |
| Listing OAM Objects | 134 |
| Using the ODL Function in Full-Screen Mode | 135 |
| Browsing and Updating OAM Objects in Full-Screen Mode | 136 |
| Printing OAM Objects | 137 |
| Controlling Print Output | 137 |
| Copying OAM Objects | 138 |
| Backing up OAM Objects in a Sequential or VSAM Data Set | 138 |
| Backing up a Single OAM Object | 138 |
| Backing up a Multiple OAM Objects | 138 |
| Restoring OAM Objects from a Sequential or VSAM Data Set | 139 |
| Restoring an Object that was Backed up Individually | 139 |
| Restoring One or More Objects from a Multiple-Object Backup | 140 |
| Erasing OAM Objects | 140 |
| Command Mode Example Using DITTO OAM Functions | 140 |
| Batch Example Using DITTO OAM Functions | 141 |
| | |
| Chapter 13. Processing Cards | 143 |
| Card Input and Output | 144 |
| Printing Card Decks | 144 |
| Controlling Print Output | 144 |
| Copying Card Decks | 145 |

Part 3. DITTO Reference 147

| | |
|--|------------|
| Chapter 14. DITTO Functions | 149 |
| How to Use this Reference Chapter | 149 |
| How to Read the Command Syntax Diagrams | 150 |
| How to Read the Batch Syntax and Keyword Syntax Diagrams | 151 |
| APB (Advanced Print Browse) | 152 |
| BS, BQ (Create Sequential Data) - MVS and VSE only | 152 |
| BSF (Backward Space File) | 156 |
| BSR (Backward Space Record) | 158 |
| BT (Create Tape File) | 159 |
| BV (Create VSAM File) | 162 |
| CC (Card to Card) | 165 |
| CCL (Card Close)—VSE Only | 165 |
| CL (Card to Library Member)—VSE Only | 166 |
| CP (Card Print) | 167 |
| CS, CQ (Card to Sequential Data)—MVS and VSE Only | 168 |

| | |
|--|-----|
| CT (Card to Tape) | 171 |
| CV (Card to VSAM) | 173 |
| DB (Disk Browse) | 175 |
| DCN (Disk to Console) | 176 |
| DID (Alter Disk VOLID)—VSE and CMS Only | 177 |
| DLA (Display Label Area)—VSE Only | 178 |
| DP (Disk Print) | 179 |
| DRL (Disk Record Load) | 181 |
| DRS (Disk Record Scan) | 182 |
| DSE (Data Set Extents) | 185 |
| DTE (Disk Track Edit) | 186 |
| DU (Disk Update) | 186 |
| DUMP (Dump CMS Files to Tape)—CMS Only | 187 |
| DVB (Data In Virtual Browse)—MVS Only | 187 |
| DVT (Display VTOC) | 188 |
| EOF (Write EOF Record) | 190 |
| EOJ (End of Job) | 190 |
| ERT (Erase Tape) | 191 |
| FB (CMS File Browse)—CMS Only | 192 |
| FL (CMS File to Library Member)—CMS/VSE Mode Only | 192 |
| FMT (Set DBCS Format) | 192 |
| FP (CMS File Print)—CMS Only | 194 |
| FS (CMS File to Sequential Data)—CMS/VSE Mode Only | 194 |
| FSF (Forward Space File) | 195 |
| FSN (Full-Screen No) | 196 |
| FSR (Forward Space Record) | 197 |
| FSY (Full-Screen Yes) | 198 |
| FT (CMS File to Tape)—CMS Only | 198 |
| FV (CMS File to VSAM)—CMS and CMS/VSE Mode Only | 200 |
| II (Interpret Cards) | 201 |
| INT (Initialize Tape) | 201 |
| LB (Library Member Browse)—VSE Only | 203 |
| LC (Library Member to Card)—VSE Only | 204 |
| LDEL (Library Member Delete)—VSE Only | 205 |
| LDL (Library Directory List)—VSE Only | 206 |
| LE (Library Member Edit)—VSE Only | 207 |
| LF (Library Member to CMS File)—CMS/VSE Mode Only | 207 |
| LL (Library Member to Library Member)—VSE Only | 208 |
| LOAD (Load CMS Files from Tape)—CMS Only | 210 |
| LP (Library Member Print)—VSE Only | 212 |
| LR (Library Member Rename)—VSE Only | 214 |
| LS (Library Member to Sequential Data)—VSE Only | 215 |
| LSL (Library Status List)—VSE Only | 217 |
| LT (Library Member to Tape)—VSE Only | 218 |
| LU (Library Member Update)—VSE Only | 219 |
| LV (Library Member to VSAM)—VSE Only | 220 |
| LVL (Display Service Level) | 221 |
| LX (Library Member to REXX Variable)—VSE only | 222 |
| MB (Memory Browse) | 223 |
| OB (Object Browse)—MVS Only | 223 |
| ODL (Object Directory List)—MVS Only | 224 |
| OE (Object Erase)—MVS Only | 226 |
| OO (Object to Object)—MVS Only | 227 |
| OP (Object Print)—MVS Only | 229 |

| | |
|--|-----|
| OS, OQ (Object to Sequential Data)—MVS Only | 230 |
| OU (Object Update)—MVS Only | 231 |
| OV (Object to VSAM)—MVS Only | 232 |
| PB (Print Browse) | 233 |
| PVT (Process VTOC)—VSE and CMS only | 234 |
| REW (Tape Rewind) | 235 |
| RUN (Tape Rewind-Unload) | 236 |
| SC, QC (Sequential Data to Card) | 237 |
| SCS (Catalog Services)—VSE and MVS Only | 239 |
| SET (Set Processing Options) | 242 |
| SF (Sequential to CMS File)—CMS and CMS/VSE Mode Only | 249 |
| SFA (Print SYSLST File Type A)—VSE Only | 251 |
| SFD (Print SYSLST File Type D)—VSE Only | 252 |
| SL (Sequential Data to Library Member)—VSE Only | 254 |
| SO, QO (Sequential Data to Object)—MVS Only | 256 |
| SP, QP (Sequential Data Print) | 258 |
| SPB (Spool Browse)—VSE and CMS Only | 260 |
| SPL (Spool List Display)—VSE Only | 260 |
| SPS (Spool Services)—VSE Only | 260 |
| SS, QQ (Sequential Data to Sequential Data)—MVS and VSE Only | 261 |
| ST, QT (Sequential Data to Tape) | 265 |
| SV, QV (Sequential Data to VSAM) | 269 |
| TB (Tape Browse) | 271 |
| TC (Tape to Card) | 272 |
| TCN (Tape to Console) | 274 |
| TF (Tape to CMS File)—CMS Only | 275 |
| TFA (Print SYSLST Tape Type A)—VSE Only | 277 |
| TFD (Print SYSLST Tape Type D)—VSE Only | 278 |
| TFT (Tape File to Tape File)—VSE and CMS Only | 279 |
| TL (Tape to Library Member)—VSE Only | 281 |
| TLB (Tape Label Display) | 283 |
| TLT (Tape to Labeled Tape) | 285 |
| TMP (Tape Map) | 287 |
| TP (Tape Print) | 289 |
| TRL (Tape Record Load) | 291 |
| TRS (Tape Record Scan) | 293 |
| TS, TQ (Tape to Sequential Data)—MVS and VSE Only | 296 |
| TT (Tape to Tape) | 300 |
| TTC (Tape to Tape Compare) | 302 |
| TTR (Tape to Tape Reblocked) | 304 |
| TU (Tape Update) | 307 |
| TV (Tape to VSAM) | 307 |
| TX (Tape to REXX Variable) | 310 |
| VB (VSAM Browse) | 312 |
| VC (VSAM to Card) | 312 |
| VDL (VSAM Data Set List) | 314 |
| VE (VSAM Edit) | 316 |
| VER (Display Service Level) | 316 |
| VF (VSAM to CMS File)—CMS and CMS/VSE Mode Only | 317 |
| VL (VSAM to Library Member)—VSE Only | 318 |
| VMC (VSAM Master Catalog Setup)—CMS Only | 320 |
| VO (VSAM to Object)—MVS Only | 321 |
| VP (VSAM Print) | 323 |
| VRU (VSAM Record Update) | 325 |

| | |
|---|-----|
| VS, VQ (VSAM to Sequential Data)—MVS and VSE Only | 326 |
| VSE (Start DITTO/ESA for VSE)—CMS Only | 329 |
| VT (VSAM to Tape) | 330 |
| VU (VSAM Update) | 333 |
| VV (VSAM to VSAM) | 334 |
| VX (VSAM to REXX Variable) | 337 |
| WTM (Write Tape Mark) | 339 |
| XL (REXX Variable to Library Member) - VSE only | 341 |
| XT (REXX Variable to Tape) | 342 |
| XV (REXX Variable to VSAM) | 344 |
| XXX (Function Summary) | 345 |

Part 4. Appendixes 347

| | |
|--|-----|
| Appendix A. Compatibility with Previous Releases of DITTO | 349 |
| DITTO Functions That are No Longer Supported | 349 |
| DITTO Function Code Synonyms | 351 |
| Batch Keywords That are No Longer Supported | 351 |
| Batch Keywords That are Not Recommended | 352 |

| | |
|-----------------------------------|-----|
| Appendix B. DITTO Messages | 353 |
| Format of Numbered Messages | 353 |

| | |
|-----------------------------------|-----|
| Notices | 373 |
| Programming Interface Information | 374 |
| Trademarks | 374 |

| | |
|-------------------------------------|-----|
| Bibliography | 375 |
| DITTO/ESA Publications | 375 |
| Related Publications for MVS | 375 |
| Related Publications for VM | 375 |
| Related Publications for VSE | 375 |
| General Publications | 375 |
| Softcopy Publications for DITTO/ESA | 376 |

| | |
|-----------------|-----|
| Glossary | 377 |
|-----------------|-----|

| | |
|--------------|-----|
| Index | 381 |
|--------------|-----|

Figures

| | | |
|-----|---|----|
| 1. | Example of Character Format | 4 |
| 2. | Example of Hexadecimal Across Format | 4 |
| 3. | Example of Hexadecimal Updown Format | 4 |
| 4. | Full-Screen Browse Functions | 5 |
| 5. | Console Display Functions | 5 |
| 6. | Full-Screen Edit and Update Functions | 6 |
| 7. | Single-Record Update Functions | 6 |
| 8. | Data Creation Functions | 6 |
| 9. | Rename Functions | 6 |
| 10. | MVS: Copy Functions | 7 |
| 11. | VSE: Copy Functions | 7 |
| 12. | CMS: Copy Functions | 7 |
| 13. | Print Functions | 8 |
| 14. | Erase Functions | 8 |
| 15. | Locate Functions | 8 |
| 16. | List Functions | 9 |
| 17. | Miscellaneous Tape Functions | 9 |
| 18. | Environment Functions | 10 |
| 19. | Task Selection Menu (VSE) | 13 |
| 20. | Primary Function Menu (MVS) | 13 |
| 21. | Primary Function Menu with Help Pull-Down Menu | 15 |
| 22. | Command Mode Example (REXX Procedure) | 23 |
| 23. | Batch Job Example: Listing DITTO Functions (MVS) | 26 |
| 24. | Batch Job Example: Listing DITTO Functions (VSE) | 26 |
| 25. | Batch Job Example: Listing DITTO Functions (CMS) | 26 |
| 26. | Full-Screen Browse Functions | 31 |
| 27. | VSAM Browse Panel | 33 |
| 28. | VSAM Browse Single-Record Panel | 35 |
| 29. | Memory Browse Panel | 36 |
| 30. | Full-Screen Edit Functions | 39 |
| 31. | VSAM Edit Panel | 40 |
| 32. | VSAM Edit Single-Record Panel | 42 |
| 33. | DITTO Functions for Catalog Processing | 51 |
| 34. | Catalog Services Selection Panel (MVS) | 53 |
| 35. | Catalog Services Selection Panel (VSE) | 53 |
| 36. | Catalog Services (Primary Panel) | 55 |
| 37. | Catalog Services (Alternative Display, MVS) | 55 |
| 38. | Functions Available for Different Types of Entries | 57 |
| 39. | Catalog Parameters Panel | 58 |
| 40. | Catalog Statistics Panel | 60 |
| 41. | DITTO Functions for Tape Processing | 61 |
| 42. | Example of Allocating Tapes in Batch Mode under MVS | 64 |
| 43. | Density/Mode Values for Tapes That Are Not 7-Track | 67 |
| 44. | Tape Conversion Mode Values for 7-Track Tapes | 67 |
| 45. | Layout of a Tape with IBM Standard Labels | 70 |
| 46. | A Tape Label Summary: Output from TLB | 72 |
| 47. | A Tape Map: Output from TMP | 73 |
| 48. | TRL Dialog Example (Line Mode) | 77 |
| 49. | Batch Example of Tape Initialization (MVS) | 83 |
| 50. | Batch Example Using DITTO Tape Functions (MVS) | 84 |

| | | |
|-----|---|-----|
| 51. | Batch Example Using DITTO Tape Functions (VSE) | 85 |
| 52. | Command Mode Example Using DITTO Tape Functions (CMS) | 85 |
| 53. | DITTO Functions for Disk Processing | 87 |
| 54. | Display VTOC | 90 |
| 55. | Display VTOC (Alternative Display) | 91 |
| 56. | Data Set Extents Display (MVS) | 92 |
| 57. | Searching for Disk Data with DRS (MVS) | 98 |
| 58. | Searching for Disk Data with DRS (VSE) | 98 |
| 59. | Command Mode Example Using DITTO Disk Functions (MVS) | 99 |
| 60. | Batch Example Using DITTO Disk Functions (MVS) | 99 |
| 61. | Batch Example Using DITTO Disk Functions (VSE) | 99 |
| 62. | DITTO Functions for Processing Sequential Data | 101 |
| 63. | Batch Example Using DITTO Sequential Data Functions (MVS) | 107 |
| 64. | Batch Example Using DITTO Sequential Data Functions (VSE) | 108 |
| 65. | DITTO Functions for Processing VSAM Data Sets | 109 |
| 66. | Batch Example Using DITTO VSAM Functions (MVS) | 115 |
| 67. | Batch Example Using DITTO VSAM Functions (VSE) | 115 |
| 68. | Command Mode Example Using DITTO VSAM Functions (CMS) | 115 |
| 69. | DITTO Functions for Processing CMS Files | 117 |
| 70. | DITTO Functions for Processing Library Members | 123 |
| 71. | Library Directory List | 125 |
| 72. | Library Directory List (Alternative Display) | 126 |
| 73. | Library Status List | 128 |
| 74. | Library Status List (Disk Extents Display) | 128 |
| 75. | Batch Example Using DITTO Library Functions | 131 |
| 76. | Command Mode Example Using DITTO Library Functions | 131 |
| 77. | DITTO Functions for Processing OAM Objects | 133 |
| 78. | Object Directory List | 135 |
| 79. | Object Directory List (Alternative Display) | 136 |
| 80. | Command Mode Example Using DITTO OAM Functions | 141 |
| 81. | Batch Example Using DITTO OAM Functions | 141 |
| 82. | DITTO Functions for Processing Cards | 143 |

About This Book

This book is intended to help you to use IBM Data Interfile Transfer, Testing, and Operations Utility for ESA (referred to in this book as DITTO/ESA or just DITTO). It contains information for system programmers and other data processing professionals who need to examine data and transfer it between storage devices.

DITTO can be used in full-screen mode, line mode, command mode, or batch mode. For full-screen mode, you can use this book as an introduction to DITTO and use the online help panels as your primary source of information. For all other modes, this book is your primary source of information.

Batch users should be familiar with job control language (JCL).

Other Sources of Information on DITTO

DITTO documentation supports the following tasks:

Evaluating DITTO: *Introducing DITTO/ESA*, GH19-8222, provides an overview of the product to help the business professional decide if DITTO has the functions needed.

| **Planning for, installing, customizing, and maintaining DITTO:** Refer first to the
| *DITTO/ESA Installation and Customization Guide*, GC26-9598, which may then
| refer you to the *Program Directory* (shipped with the product tape) for some
| information.

Using DITTO: This book, the *DITTO/ESA User's Guide and Reference*, is a guide to using DITTO. You can also refer to:

- Reference summary
DITTO/ESA Reference Summary, SX11-6110 summarizes the DITTO functions.
- Function listing
The XXX function produces a list of DITTO functions and parameters.
- Online help
In full-screen mode, online help is available.

| The latest DITTO documentation and product news is available on the Web at:
| <http://www.software.ibm.com/ad/ditto/>

| For information about related products, see the "Bibliography" on page 375.

Summary of Changes

This section lists the major changes that have been made to the DITTO/ESA product and this book. Technical changes are marked in the text by a change bar in the left margin.

Major Changes to DITTO/ESA

Release 3, June 1999

- The following enhancements improve tape support:
 - For 3590 devices, you can now limit the amount of data stored on a tape to a percentage of its capacity, using the new SCALE parameter of the Initialize Tape (INT) function.
 - Standard Label (SL) processing for basic tape functions under MVS.
 - Support for ANSI Version 4 tape labels.
- VSE External Security Manager support offers improved security checking for VSE users.
- The new Spool Services (SPS) function enables VSE users to work with all VSE POWER spool queues (LST, PUN, RDR, XMT) using DITTO/ESA's full-screen interactive user interface. This function lists all entries of the selected queue and allows you to enter a command directly on an entry to browse, alter, delete, hold, release, or transfer it.
- Support for wide screens in DITTO/ESA list functions, displaying more information without scrolling.
- Improved performance under MVS of System Catalog Services (SCS) and VSAM Data Set List (VDL) functions, by exploiting the new catalog search interface introduced by DFSMS™ 1.4. The full-screen display now includes more SMS-related information, and statistical information has changed from byte to kilobyte (KB) to support larger amounts.
- In full-screen mode, enhanced dynamic QSAM data set allocation allows you to specify an existing data set to be used as a model for the new data set (LIKE option) and to release any unused space when the data set is closed (RELEASE option).
- You can now sort the output of the Library Directory List (LDL) function by library name then member name within each library, using the new LIB value of the SORTBY parameter.
- In full-screen mode, the new Tab-to-action-bar option on the View menu changes the behavior of the Tab key so that it moves the cursor to the action bar.
- In full-screen mode, the new ABAR command displays or hides the action bar.
- When editing data in full-screen mode, you can use the new ISPF-like CAPS command to translate text into uppercase. The CAPS command is similar to, but not identical to, the existing CASE command.

- For improved compatibility with the ISPF editor, you can specify the syntax of the browse and edit LOCATE subcommand and the edit CHANGE subcommands, using the new CMDSYNTAX parameter of the SET function.
- The Display VTOC (DVT) function now displays the record format, record size, and block size values from the new VTOC Format 1 label.
- In full-screen mode, the Catalog Services (SCS) function now displays extent information for VSAM files.
- New MOVE and REPLACE parameters in the Library Member to Library Member (LL) function allow you to move and optionally replace VSE Librarian members.
- If DFSMS/MVS 1.4 or later is installed on your system, the Catalog Services (SCS) function now lists all catalog entries when you specify a data set name of '***'. Previously, the SCS function listed only the master catalog entries.
- Functions that were listed in the *DITTO/ESA Release 2 User's Guide and Reference* Appendix A, "Compatibility with Previous Releases of DITTO," under the heading "DITTO Functions That are Not Recommended," are no longer supported in Release 3.

Release 2, December 1996

- Year 2000 support, to sort and display dates beyond 2000.
- A new Advanced Print Browse (APB) function provides a readable display to quickly browse Advanced Function Printing™ (AFP™, LIST3820) format data.
- VM/VSE users can now execute DITTO/ESA for VSE on a local or remote VSE system directly from a VM DITTO/ESA session in client/server mode. In this mode, all of the normal VSE functions are available as well as additional functions to copy CMS files to and from library members, SAM files, and VSAM files.
- A new function, DLA, displays or prints the contents of the VSE label area. From the DLA display, DITTO line commands can be entered to work with the listed data sets.
- A new Disk Track Edit (DTE) function. This function lets you insert and delete records, and change the length and contents of key and data on a Count Key Data (CKD) disk track.
- New functions (LS, SL) for copying VSE library members to and from sequential data sets.
- Padding/truncating records while copying.
- Additional record formats, spanned records for tape.
- Enhanced TLT function to better support continuation volumes.
- Improved tape end-of-volume processing.
- Support for the IBM 3590 tape drive.
- ANSI label support.
- Performance enhancements for various functions.
- DITTO/ESA is now enabled for NLS translation.

Release 1, April 1995

DITTO/ESA is a follow-on product for the MVS/DITTO product, the DITTO for VSE and VM base product, and the DITTO 3.2 Productivity Features.

Changes for OS/390 or MVS Users

DITTO/ESA provides the following enhancements:

- Exploitation of MVS/ESA™
- Task-oriented function selection panels
- The ability to use keywords in command mode
- Support for disk fullpack processing in batch mode
- Enhanced security
- Enhanced tape bypass label processing (BLP) support
- The ability to specify a panel library, user profile data set, and OAM database name at customization time
- A new VSAM Edit function
- New functions to copy REXX stem variables to and from tape files and VSAM data sets
- A function to display disk information on the console
- Enhanced disk browse and update support for new MVS data structures
- Additional statistical information about catalogs
- Object Access Method (OAM) enhancements
- Enhanced Tape to Labeled Tape function
- The ability to route print output to a REXX stem variable
- Additional print format option for the TMP function
- The ability to set a tape mode code in batch mode
- Double-byte character set (DBCS) options
- The ability to copy a multifile tape to a sequential data set
- Support for tape blocks longer than 64KB
- New fill patterns for the data creation functions

Changes for VSE Users

DITTO/ESA provides the following enhancements:

- Exploitation of VSE/ESA™
- Task-oriented function selection panels
- The ability to use keywords in command mode
- Automated tape library support
- Support for some DITTO functions in the VSE stand-alone environment
- A new VSAM Edit function
- New Library Member Edit and Library Member Rename functions

- New functions to copy REXX stem variables to and from tape files, VSAM data sets, and library members
- A function to display disk data set extents
- Enhanced Tape to Labeled Tape function
- The ability to copy, rename, and delete multiple library members
- The ability to limit a library list by creation date
- The ability to set a tape mode code in batch mode
- The ability to set the page size for print output
- The ability to limit a catalog list or VSAM list by space ID
- The ability to copy a multifile tape to a sequential data set
- Additional statistical information about catalogs and library members
- New fill patterns for the data creation functions

The following enhancements are incorporated from the DITTO 3.2 Productivity Features:

- Enhanced Common User Access® (CUA®) interface, with more extensive online help
- VSE/ICCF is no longer required to run DITTO in full-screen mode under VSE. Full-screen DITTO is now started as a CICS® transaction under VSE.
- Enhanced security, through the use of a security exit
- A connection to REXX/VSE, which allows easy integration of DITTO commands in user-defined procedures
- A function for copying multifile, multivolume standard labeled tapes
- A function for working with lists of members in a library
- Browse, update, print, and erase functions for library members
- Functions for copying library members to and from tape, VSAM files, and cards
- A function for working with lists of catalog entries. This function also lets you display, define, delete, and alter VSAM catalog entries
- The ability to route print output to a REXX stem variable
- The ability to assign tapes permanently (for the duration of a DITTO session)
- Batch support for the ERT and PVT functions
- Additional print format option for the TMP function
- Improved tape record format support
- Improved tape and DASD error handling
- Support for tape blocks longer than 64KB

Changes for CMS Users

DITTO/ESA provides the following enhancements:

- Exploitation of VM/ESA®
- Task-oriented function selection panels
- The ability to use keywords in command mode
- Support for VSAM input, VSAM output, and SAM input functions in batch mode
- A new VSAM Edit function
- New functions to copy REXX stem variables to and from tape files and VSAM data sets
- A function to display disk data set extents

- Enhanced Tape to Labeled Tape function
- The ability to set a tape mode code in batch mode
- The ability to set the page size for print output
- The ability to limit a VSAM list by space ID
- New fill patterns for the data creation functions

The following enhancements are incorporated from the DITTO 3.2 Productivity Features:

- Enhanced Common User Access (CUA) interface, with more extensive online help
- Enhanced security, through the use of RACF® or a security exit
- Support for labeled tape processing in batch mode
- A function for copying multifile, multivolume standard labeled tapes
- The ability to route print output to a REXX stem variable
- Batch support for the ERT and PVT functions
- Additional print format option for the TMP function
- Improved tape record format support
- Improved tape and DASD error handling
- Support for tape blocks longer than 64KB

Compatibility with Previous Releases of DITTO

Several parameters have been changed in DITTO/ESA. For compatibility, most of the old parameters are still supported in batch mode. In interactive mode, several commands can be entered in a shorter form.

In line mode and command mode, many commands have new parameters or have the same parameters in a different order. To run an existing procedure with DITTO/ESA, you should compare all of the DITTO commands with the syntax described in this book.

If you want a procedure to run unchanged from one release of DITTO to the next, you should use the new keyword syntax for all DITTO commands in the procedure. Any DITTO commands that are specified with keywords are not affected by changes in the prompting order for parameters.

The following functions are *not supported* in DITTO/ESA:

- Functions that work with ISAM files
- Functions that work with diskettes
- Functions that work with 96-column cards
- Functions that work with split-cylinder disk data
- Invoking full-screen mode DITTO under VSE/ICCF.

For full details of compatibility with previous releases, refer to Appendix A, “Compatibility with Previous Releases of DITTO” on page 349.

Changes to This Book

Third Edition (June 1999)

The following change has been made to this book since the second edition:

- Appendix B, "Installing and Customizing DITTO," and Appendix C, "Customizing the Security Environment," have been removed. This information is presented in the new *DITTO/ESA Installation and Customization Guide*, GC26-9598.

Part 1. Introduction to DITTO

For users new to DITTO

Part 1, "Introduction to DITTO" gives an overview of DITTO functions, and the different ways you can use DITTO.

For users already familiar with DITTO basics

Part 2, "Guide to Using DITTO Functions" on page 29 describes what you can accomplish with DITTO functions. The first two chapters describe browse and edit functions that are available in full-screen mode. The remaining chapters are divided according to the type of input you are working with: catalog entries, tapes, disk records, sequential data (QSAM or SAM), VSAM data, CMS files, VSE library members, OAM objects, or cards.

For experienced DITTO users

Part 3, "DITTO Reference" on page 147 presents the DITTO functions in alphabetical order.

For special situations

Part 4, "Appendixes" on page 347 contains information on compatibility with previous releases, installing DITTO, controlling DITTO security, and responding to DITTO error messages.

If you are running DITTO in full-screen mode, you can also refer to the online help.

Introduction

Chapter 1. Overview of DITTO Functions

This chapter gives an overview of the functions provided by DITTO. These functions are described in more detail in Part 2, “Guide to Using DITTO Functions” on page 29.

Where You Can Run DITTO

DITTO/ESA Release 3 runs in the following environments:

- Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA) SP Version 5 Release 2
- Virtual Machine/Enterprise Systems Architecture (VM/ESA) Version 2
- Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA) Version 2 Release 4

or subsequent releases, unless otherwise stated. For details, see the *DITTO/ESA Installation and Customization Guide*, GC26-9598.

What Types of Data You Can Work With

You can work with the following types of data:

- Physical tape records (blocks)
- Physical disk records (blocks)
- Punched cards (or card-image data on tape or disk)
- Sequential data (SAM or QSAM)
- VSAM data
- OAM objects under MVS
- CMS files
- DDR backup tape of CMS minidisks
- Library members under VSE
- REXX stem variables
- Print output
- User storage.

What You Can Do With Data

You can perform the following tasks:

- Look at data
- Find data
- Change data
- Copy data
- Print data
- Erase data
- Work with a list of data sets, objects, or members
- Position a tape
- Compare two tapes
- Initialize or erase a tape
- Query and change your processing environment.

These general tasks are described in more detail in the remainder of this chapter.

Formats for Displaying or Printing Data

You can display or print data in three formats:

- Character format
- Hexadecimal across format (system dump format)
- Hexadecimal updown format (DITTO dump format)

The following figures show a single record in each format.

```
REC      1 DATA  133      ANDREW ASTLE * TYRONE DALAIS * GRANT SUTHE
                                RLAND * KEITH STEWART * GRAHAM HANNINGTON
                                * JOHN LEVERINGTON * LIZ RUSHTON * BOB
```

Figure 1. Example of Character Format

```
CYL-HD-REC 00163-03-001 DATA LEN  133
000000 C1D5C4D9 C5E640C1 E2E3D3C5 405C40E3 *ANDREW ASTLE * T*
000010 E8D9D6D5 C540C4C1 D3C1C9E2 405C40C7 *YRONE DALAIS * G*
000020 D9C1D5E3 40E2E4E3 C8C5D9D3 C1D5C440 *RANT SUTHERLAND *
000030 5C40D2C5 C9E3C840 E2E3C5E6 C1D9E340 ** KEITH STEWART *
000040 5C40C7D9 C1C8C1D4 40C8C1D5 D5C9D5C7 ** GRAHAM HANNING*
000050 E3D6D540 5C40D1D6 C8D540D3 C5E5C5D9 *TON * JOHN LEVER*
000060 C9D5C7E3 D6D5405C 40D3C9E9 40D9E4E2 *INGTON * LIZ RUS*
000070 C8E3D6D5 405C40C2 D6C24040 40404040 *HTON * BOB *
000080 40404040 40 * *
```

Figure 2. Example of Hexadecimal Across Format

```
CYL  163 DATA  133 CHAR ANDREW ASTLE * TYRONE DALAIS * GRANT SUTHERLAND *
HEAD  3          ZONE CDCDCE4CEEDC454EEDDDC4CCDCCE454CDCDE4EECCDDCDC454
REC   1          NUMR 1549560123350C038965504131920C079153024385931540C0
                   01...5...10...15...20...25...30...35...40...45...50
CHAR KEITH STEWART * GRAHAM HANNINGTON * JOHN LEVERINGT
ZONE DCCEC4EECECDE454DCDCDD4CCDDCDEDD454DDCD4DCECDCDCE
NUMR 25938023561930C0791814081559573650C016850355599573
                   51...55...60...65...70...75...80...85...90...95.....
CHAR ON * LIZ RUSHTON * BOB
ZONE DD454DCE4DEECEDD454CDC44444444444
NUMR 650C0399094283650C026200000000000
                   101...5...10...15...20...25...30...
```

Figure 3. Example of Hexadecimal Updown Format

Use the DUMP keyword of the SET command to specify which hexadecimal format (across or updown) is used.

You can also print double-byte character set (DBCS) data, if you have the necessary hardware. Use the FMT command to specify which data in each record is DBCS, and use the DBCSPRT keyword of the SET command to specify how DBCS data is printed.

If you want to align DBCS and EBCDIC data in printed output, use a font that sets the width of each DBCS character to 2 EBCDIC bytes.

Looking at Data

DITTO provides several full-screen browse functions that let you look at data without changing it. These functions are shown in Figure 4. To view records from line mode at the MVS or VSE system console (where full-screen browse functions are not available), you can use the functions shown in Figure 5.

Figure 4. Full-Screen Browse Functions

| What You Browse | Function |
|---------------------------|----------|
| Tape records | TB |
| Disk records | DB |
| Print output | APB, PB |
| User storage | MB |
| VSAM data | VB |
| MVS: VSAM data-in-virtual | DVB |
| MVS: OAM objects | OB |
| VSE: library members | LB |
| VSE and CMS: Spool files | SPB |
| CMS: CMS files | FB |

Within a browse function, you can use subcommands to:

- Move in any direction
- Locate a specific string
- Print the current screen image
- Put data into a buffer for later use
- Purge data (in the **PB** and **SPB** functions only)

DITTO's browse functions are very efficient, because they read only the data for the current display (one screen of data) into storage.

Figure 5. Console Display Functions

| What You Display | Function |
|------------------|----------|
| Tape records | TCN |
| Disk records | DCN |

Note: If you enter the functions listed in Figure 5 from full-screen mode, corresponding full-screen functions (**TB** and **DB**) are invoked instead.

Changing Data

DITTO provides several full-screen edit and update functions that let you change data. These functions are shown in Figure 6. If full-screen mode is not available (for example, at the system console or within a batch job), you can use the functions shown in Figure 7.

In addition, DITTO provides functions that let you create new data and rename existing entities. These functions are shown in Figure 8 and Figure 9.

Figure 6. Full-Screen Edit and Update Functions

| What You Update | Function |
|---------------------|----------|
| Tape records | TU |
| Disk records | DTE, DU |
| VSAM data | VU, VE |
| MVS: OAM object | OU |
| VSE: library member | LU, LE |

Within a full-screen update function, you can use subcommands to:

- Move in any direction
- Locate a specific string
- Print the current screen image
- Put data into a buffer for later use
- Get data from the buffer
- Copy data within a record, member, or object.

Figure 7. Single-Record Update Functions

| What You Update | Function |
|-----------------|----------|
| Tape records | TRL |
| Disk records | DRL, EOF |
| VSAM data | VRU |

Figure 8. Data Creation Functions

| What You Create | Function |
|-----------------------------|----------|
| Tape records | BT |
| VSAM data | BV |
| MVS or VSE: Sequential data | BS, BQ |

Figure 9. Rename Functions

| What You Rename | Function |
|-------------------------------|----------|
| VSE or CMS: disk VTOC entries | PVT |
| VSE or CMS: disk volume ID | DID |
| VSE: library members | LR |

Copying Data

DITTO provides several functions that let you copy data. These functions are shown in Figure 10, Figure 11, and Figure 12.

Figure 10. MVS: Copy Functions

| Copy From | Copy To | | | | | |
|--------------|--------------|--------|--------|--------|--------|------|
| | Tape | VSAM | QSAM | OAM | Cards | REXX |
| Tape | TT, TTR, TLT | TV | TS, TQ | - | TC | TX |
| VSAM | VT | VV | VS, VQ | VO | VC | VX |
| QSAM | ST, QT | SV, QV | SS, QQ | SO, QO | SC, QC | - |
| OAM | - | OV | OS, OQ | OO | - | - |
| Cards | CT | CV | CS, CQ | - | CC, II | - |
| REXX | XT | XV | - | - | - | - |

Figure 11. VSE: Copy Functions

| Copy From | Copy To | | | | | |
|----------------|-------------------|--------|--------|---------|--------|------|
| | Tape | VSAM | SAM | Library | Cards | REXX |
| Tape | TT, TTR, TFT, TLT | TV | TS, TQ | TL | TC | TX |
| VSAM | VT | VV | VS, VQ | VL | VC | VX |
| SAM | ST, QT | SV, QV | SS, QQ | SL | SC, QC | - |
| Library | LT | LV | LS | LL | LC | LX |
| Cards | CT | CV | CS, CQ | CL | CC, II | - |
| REXX | XT | XV | - | XL | - | - |

If you start full-screen VSE/DITTO from CMS (CMS/VSE mode), you can also copy CMS files to and from library members (FL, LF), SAM files (FS, SF), and VSAM files (FV, VF).

Figure 12. CMS: Copy Functions

| Copy From | Copy To | | | | |
|--------------|-------------------|--------|----------|--------|------|
| | Tape | VSAM | CMS | Cards | REXX |
| Tape | TT, TTR, TFT, TLT | TV | LOAD, TF | TC | TX |
| VSAM | VT | VV | VF | VC | VX |
| SAM | ST, QT | SV, QV | SF | SC, QC | - |
| CMS | DUMP, FT | FV | - | - | - |
| Cards | CT | CV | - | CC, II | - |
| REXX | XT | XV | - | - | - |

Printing Data

DITTO provides several functions that let you print data. These functions are shown in Figure 13.

| <i>Figure 13. Print Functions</i> | |
|-----------------------------------|-----------------|
| What You Print | Function |
| Tape records | TP |
| Disk records | DP |
| Cards | CP |
| Sequential data | SP, QP |
| VSAM data | VP |
| MVS: OAM object | OP |
| VSE: library member | LP |
| VSE: SYSLST disk file | SFA, SFD |
| VSE: SYSLST tape file | TFA, TFD |
| CMS: CMS file | FP |

You can browse print output using the **PB** function.

Erasing Data

DITTO provides functions that let you erase data. These functions are shown in Figure 14.

| <i>Figure 14. Erase Functions</i> | |
|-----------------------------------|-----------------|
| What You Erase | Function |
| Tape records | ERT |
| MVS: OAM object | OE |
| VSE: library member | LDEL |
| VSE or CMS: disk VTOC entry | PVT |

Finding Data

In most cases, using a full-screen browse function is the easiest way to locate data. If full-screen mode is not available (for example, at the system console or within a batch job), you can use the **TRS** and **DRS** functions. You can use the **DSE** function to find the physical location of a data set on a OS (MVS) or DOS (VSE) formatted disk.

| <i>Figure 15. Locate Functions</i> | |
|------------------------------------|-----------------|
| What You Locate | Function |
| Tape records | TRS |
| Disk records | DRS |
| Disk data set | DSE |

Working with a List of Data Sets, Members, or Objects

DITTO provides several functions that let you work with lists of data sets, members, or objects. These functions are shown in Figure 16.

Figure 16. List Functions

| What You List | Function |
|-----------------------------|-----------------|
| Disk contents (VTOC) | DVT |
| VSAM data | VDL |
| MVS: OAM objects | ODL |
| MVS or VSE: catalog entries | SCS |
| VSE: library members | LDL |
| VSE: library status | LSL |
| VSE: label area | DLA |
| VSE: POWER queue entries | SPL, SPS |

Within each list, you can select individual data sets, members, or objects to browse or update.

Working with Tapes

DITTO provides several functions that let you work with tapes. Those tape functions that are not shown elsewhere in this chapter are shown in Figure 17.

Figure 17. Miscellaneous Tape Functions

| What You Do | Function |
|----------------------------------|-----------------|
| Space forward by file or record | FSF, FSR |
| Space backward by file or record | BSF, BSR |
| Rewind to beginning of tape | REW |
| Rewind and unload | RUN |
| Initialize | INT |
| Write tape marks | WTM |
| Compare two tapes | TTC |
| Print tape labels | TLB |
| Print a tape map | TMP |

Querying and Changing Your Processing Environment

DITTO provides several functions that let you query or change your processing environment. These functions are shown in Figure 18.

Figure 18. Environment Functions

| What You Do | Function |
|---------------------------------|----------|
| Query DITTO version | LVL, VER |
| List DITTO functions and syntax | XXX |
| Switch execution mode | FSY, FSN |
| Invoke CMS/VSE mode | VSE |
| Format DBCS data | FMT |
| End DITTO job or session | EOJ |
| Change processing options | SET |
| VSE: cancel card input | CCL |
| CMS: set VSAM master catalog | VMC |

With the **SET** function, you can control:

- What happens when you use a print function:
 - Where the output is sent
 - How many columns wide the output is
 - Which part of each record the output contains
 - How many lines per page the output has
 - Whether output from each function starts on a new page
 - Whether the output contains a header page at the beginning
 - Whether the output contains a data header for each record
 - Whether non-printable characters are translated to blanks
 - How double-byte characters are printed
 - Which format (updown or across) hexadecimal output has
- What delimiter marks the end of tape or card input
- Whether data is translated between ASCII and EBCDIC formats
- Whether ANSI or EBCDIC labels are created
- Whether records are padded or truncated while being copied
- Which language is used for DITTO panels and messages (if translated)
- The syntax of the browse and edit LOCATE command and the edit CHANGE command, for compatibility with different platforms
- MVS: whether you will specify absolute or relative disk addresses
- VSE: whether tapes are assigned temporarily or permanently.

Chapter 2. Running DITTO in Different Modes

DITTO is available in four modes:

- Full-screen** In full-screen mode, you can use either menus or commands to select functions. If a function has parameters, you can enter them on a full-screen panel. Online help is available at all times.
- Line** In line mode, you enter commands with or without parameters. If you do not supply parameters, you are prompted for them in “question and answer” style. After a command is executed, you remain in DITTO and can enter additional commands. You can use line mode at a system console that does not support full-screen interaction.
- Command** In command mode, you enter a single command with all appropriate parameters from outside DITTO (for example, from a selection list or within a procedure). You can use either the parameter syntax (similar to line mode) or the keyword syntax (similar to batch mode). The command invokes DITTO, and you automatically leave DITTO after the command is executed.
- Batch** In batch mode, you code DITTO commands as part of a batch job. You must be familiar with JCL. Some DITTO functions, especially those that print the contents of a disk or tape, are ideally suited to batch mode.

Running DITTO in Full-Screen Mode

In full-screen mode, you can use menus, online help, and interactive browse and update functions. You will probably find full-screen mode the most convenient way to run DITTO, especially if you are a new DITTO user.

MVS or CMS: Starting DITTO in Full-Screen Mode

To start DITTO in full-screen mode under MVS or CMS, enter:

```
DITTO [function]
```

from a TSO or CMS terminal.

Note: Under MVS, the procedure for starting DITTO may depend on whether you are running DITTO APF-authorized. To find out which start procedure you should use, talk to your system administrator. For information about the implications of running DITTO APF-authorized, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

The first time you use DITTO, a copyright panel appears. After reading the panel, press the Cancel function key. In future DITTO sessions, this panel will not automatically appear.

If you do not specify a DITTO function, the DITTO Task Selection menu appears. The menu for MVS and CMS is similar to the one shown for VSE in Figure 19 on page 13.

If you would prefer to work from a list of functions, select F2 from the Task Selection menu to display the Primary Function Menu. This menu is shown in Figure 20 on page 13.

Modes

To start DITTO without going through the Task Selection menu, enter DITTO followed by either a function name or a selection panel number. For example, to start DITTO and go directly to the Set Processing Parameters panel, enter either of these commands:

```
DITTO SET
DITTO 13
```

VSE: Starting DITTO in Full-Screen Mode

Under VSE, full-screen DITTO is started and controlled by a CICS transaction and runs in a dynamic or standard VSE partition. During installation and customization, you set the default job parameters for the batch partition.

To start DITTO in full-screen mode under VSE, enter:

```
DITTO [function] [(options)]
```

from a CICS terminal, where:

function Is an optional function name or menu selection. This lets you skip the Task Selection menu and go directly to the panel for the function that you want.

options Override the following default job parameters:

| | |
|------------------------|--|
| JOBCLASS=c | The job class for DITTO |
| ACCOUNT=account | The job account information |
| WAIT=nnnn | The timeout value (in seconds) to limit waiting for connection to the DITTO batch partition. The range is 1 to 9999; the default is 120. |
| USER=userid | The user ID |
| PWD=pwd | The password |
| LSTCLASS=c | The class for DITTO print output |
| LSTDISP=d | The disposition for DITTO print output |
| TOUSER=user | The destination user ID for DITTO print output |
| PUNCLASS=c | The class for DITTO punch output |
| PUNDISP=p | The disposition for DITTO punch output |
| OPTION=options | Parameters for the OPTION JCL statement separated by commas (no blanks allowed) |

For example, you could invoke full-screen DITTO under VSE in any of the following ways:

```
DITTO
DITTO SET
DITTO 13 (USER=DITUSR01 PWD=X3AA2PQ
DITTO (JOBCLASS=Z LSTDISP=D TOUSER=KEISTEW
DITTO (OPTION=PARTDUMP,NOSYSDDUMP
```

The first time you use DITTO, a copyright panel appears. After reading the panel, press the Cancel function key. In future DITTO sessions, this panel will not automatically appear.

If you do not specify a DITTO function, the Task Selection menu appears. The VSE version of the Task Selection menu is shown in Figure 19 on page 13.

```

Process  View  Options  Help
-----
DITTO/ESA for VSE          Task Selection Menu

Select the desired task or enter a DITTO function code, then press Enter.
Use the Menu key to display the menu panel with DITTO function groups.

-----
  1. Browse data
  2. Edit or update data
  3. Work with VTOC
  4. Work with VSAM catalog
  5. Work with VSE libraries
  6. Print data
  7. Copy data
  8. Locate data
  9. Change data
 10. Create data
 11. Position a tape
 12. Tape specific functions
 13. Set processing options

Command ==>
F1=Help F2=Menu F3=Exit F10=Actions F11=CRetrieve F12=Cancel

```

Figure 19. Task Selection Menu (VSE)

If you would prefer to work from a list of functions, select F2 from the Task Selection Menu to display the Primary Function Menu. The MVS version of the Primary Function menu is shown in Figure 20.

```

Process  View  Options  Help
-----
DITTO/ESA for MVS          Primary Function Menu

Select the desired function group or enter a DITTO function code, then press
Enter. Press F2 (Menu) to get task-oriented guidance to DITTO functions.

-----
  0. CONTROL          - DITTO control functions and tools
  1. CARD              - Card functions
  2. TAPE CONTROL     - Tape control functions
  3. TAPE              - Tape functions
  4. DISK              - Disk functions
  5. QSAM              - Sequential data functions
  6. VSAM              - VSAM functions
  7. OAM               - OAM functions
  8. SCS               - System catalog services
  9. XXX LISTING      - Print DITTO function summary
 10. NEWS              - Display DITTO release news
 11. EOJ               - Terminate DITTO program

Command ==>
F1=Help F2=Menu F3=Exit F10=Actions F11=CRetrieve F12=Cancel

```

Figure 20. Primary Function Menu (MVS)

CMS/VSE Mode: Starting VSE/DITTO from CMS

Under CMS, you can invoke a full-screen VSE/DITTO session in client server mode (CMS/VSE mode). Within this session, you can use any function that is available to VSE/DITTO users. Additionally, you can copy CMS files to and from the following:

- Library members (FL and LF functions)
- SAM files (FS and SF functions)
- VSAM files (FV and VF functions).

In client server mode, all data access (SAM and VSAM) is done under the control of VSE rather than CMS/DOS and CMS/VSAM. There is no need to link to the VSE disks. This provides full integrity and access protection for catalogs and data as defined for the VSE system.

A VM user must be enabled for the CMS/VSE mode session, as described in the *DITTO/ESA Installation and Customization Guide*.

To start VSE/DITTO in full-screen mode under CMS, either select VSE on the CMS/DITTO main menu or enter:

```
DITTOVSE [function] [(options)]
```

function Is an optional function name or menu selection. This lets you skip the Task Selection menu and go directly to the panel for the function that you want.

options Override the following default job parameters:

| | |
|------------------------|---|
| USER userid | The VSE security user ID |
| PWD pwd | The password. If a question mark (?) is specified, DITTO prompts for the password using non-displayed input. |
| VSE user | The VSE system user ID, optionally followed by a node in the form userid[@node] or userid[AT node]. |
| WAIT=nnnn | The timeout value (in seconds) to limit waiting for connection to the DITTO batch partition. The range is 1 to 9999; the default is 120. |
| TOUSER user | The destination for the DITTO print and punch output. Specify a user ID, optionally followed by a node in the form userid[@node] or userid[AT node]. |
| DIST dist | The distribution code for DITTO print and punch output |
| JOBCLASS c | The VSE job class for DITTO |
| ACCOUNT account | The VSE job account information |
| LSTCLASS c | The class for DITTO print output |
| LSTDISP d | The disposition for DITTO print output |
| PUNCLASS c | The class for DITTO punch output |
| PUNDISP p | The disposition for DITTO punch output |
| OPTION options | Parameters for the OPTION JCL statement |

For example:

```
DITTOVSE (USER abc123 PWD ? VSE vse1@xycity
```

In CMS/VSE mode, type CMS or CP before any CMS or CP command that you want to execute on your CMS user's virtual machine.

The DITTO Full-Screen Panels

DITTO's panels are based on the Common User Access (CUA) guidelines. The fundamental objectives of CUA are:

- Usability and consistency within an application
- Consistency between applications

If you have used other CUA products, you should find the DITTO panels familiar.

An *action bar* at the top of each CUA panel lets you select various choices. To move the cursor to the action bar, use the tab and arrow keys or press the Actions function key. To select a choice from the action bar, move the cursor to the choice, then press Enter.

When you select an action bar choice, a pull-down menu appears below the action bar. Figure 21 shows a DITTO panel with the Help pull-down menu open. To select a choice, type the number corresponding to the choice or use the tab and arrow keys to move the cursor to the choice, then press Enter.

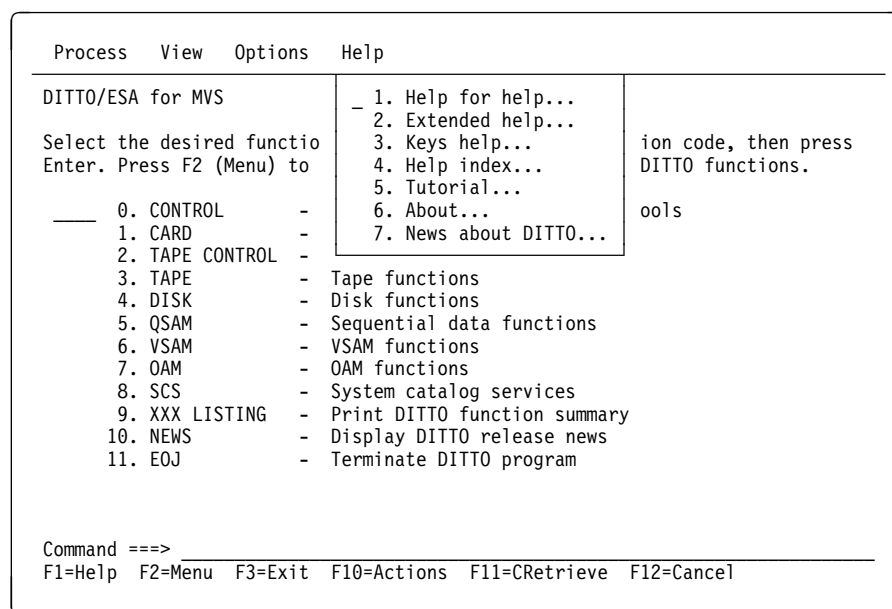


Figure 21. Primary Function Menu with Help Pull-Down Menu

Online Help

These types of online help are available:

Field help Specific help information about the item that the cursor is on. Field help is available for input fields on panels, and also for certain highlighted phrases within the help text. To get field help, use the tab key or arrow keys to move the cursor to the appropriate item, then press the Help function key.

Help for help Information about how to use the online help facility.

| | |
|----------------------|---|
| Extended help | Help information about an entire function or panel. There are three ways to get extended help: <ul style="list-style-type: none">• Select Extended help from the Help pull-down menu.• Move the cursor to a position where no field help is available, and press the Help function key.• Press the Extended help function key from within field help. |
| Keys help | A list of the keys that are available in DITTO, and their functions. |
| Help index | A list of help information for DITTO. |
| Copyright | To get copyright information for this release of DITTO, select About from the Help pull-down menu. |
| News | A list of new features in the current release of DITTO. To get news, select News about DITTO from the Help pull-down menu. |
| Message help | More information about a message. To get message help, press the Help function key when a message with a number in the form <code>DITnnnnx</code> appears. Message help appears once, regardless of the cursor position. If you later want to redisplay the message help, move the cursor to the message line and press the Help function key. |
| Tutorial | A general introduction to the tasks you can perform in full-screen mode DITTO. To get the tutorial, press the Tutorial function key from within the online help. |

To learn more about the help facility, try pressing the Help function key in various situations.

Function Keys

The function keys that are available on a DITTO panel are listed at the bottom of the panel. The following function keys are available on most DITTO panels:

| | |
|----------------------|--|
| F1=Help | Displays help information appropriate for the current cursor position. |
| F3=Exit | Exits the current function or panel. |
| F6=Rfind | Repeat the previous FIND command. |
| F7=Bkwd | Moves backward through the displayed data. |
| F8=Fwd | Moves forward through the displayed data. |
| F10=Actions | Moves the cursor to the action bar. |
| F11=CRetrieve | Retrieves the previous command that you entered (if the cursor is on the command line; if not, it returns the cursor to the command line). |
| F12=Cancel | Ignores anything that you typed on the current panel, and leaves the current panel. |

On some DITTO panels, additional function keys are available. For information about what a function key does, see the keys help online.

Enter **KEYS** to set the values of function keys 13 to 24. On the panel that appears, you can also specify that the terminal type is 3270KN if you are using a terminal that supports Japanese Katakana characters.

If you specify 3270KN, DITTO will translate all message text to uppercase if **SET LANGUAGE=ENGLISH** is specified, and will translate panel text to uppercase regardless of the language.

Selecting a Function

To select a DITTO function from the Primary Function menu, you can:

- Move the cursor to the function you require, and press Enter. For example, move the cursor to 0 on the Primary Function Menu and press Enter, then move the cursor to 1 on the panel that is displayed and press Enter again.
- Enter the function name. For example, enter **SET** on the command line.
- Enter a selection number. For example, enter 0 on the command line, then enter 1 on the command line of the panel that appears.
- Enter a composite selection number. For example, enter 0.1 on the command line.
- Enter a composite selection number with an equal sign in front. For example, enter =0.1 on the command line.

Whichever method you choose, the appropriate panel (in the examples above, the Set Processing Parameters panel) appears.

The following functions are available only in full-screen mode:

- Full-screen browse, edit, and update functions.
- Online help and information panels.
- The following functions, which are available on the command line of any DITTO panel:

DX Converts a number from decimal to hexadecimal. For example, dx 16 displays the value 10.

XD Converts a number from hexadecimal to decimal. For example, xd 10 displays the value 16.

KEYS Lets you define the settings of function keys 13 to 24, or specify that you are using a Katakana terminal.

PRINT Prints the current screen image.

PFSHOW Turns on or off the display of function keys at the bottom of the screen. By default, function keys are displayed.

PANELID Turns on or off the display of panel names at the top of the screen. By default, panel names are not displayed.

CMDLINE Sets the position of the command line (top or bottom of the panel).

- Under MVS, you can enter a TSO command on most DITTO panels. Type **TSO** before the command name.
- Under CMS, you can enter a CP or CMS command on most DITTO panels. If this command has the same name as a DITTO command, type **CP** or **CMS** before the command name.

MVS: Running DITTO under ISPF

If you are running DITTO in non-authorized mode under ISPF, F9=Swap is listed at the bottom of DITTO panels. This function key lets you swap back and forth between DITTO and the ISPF session.

DITTO panels always use the entire screen, even in ISPF split-screen mode. For this reason, the behavior of the Swap function key might not be exactly what you expect.

To get the best results, you should force the screen to split at line 1 (the line at the top of the panel). There are two ways to do this:

- If you have not started the DITTO session yet, move the cursor to line 1 and press the ISPF SPLIT function key. On the split screen that appears, enter TSO DITTO.
- If you have already started the DITTO session, press the Swap function key. On the panel that appears, move the cursor to line 1 and press the ISPF SPLIT function key.

The Swap function key (or the ISPF SWAP command) will now move you back and forth between a full-screen DITTO panel and an ISPF panel.

Note: If you enter DITTO from an ISPF panel that has the command line at the bottom, you might need to press the DITTO Swap key twice before the actual swap occurs.

CMS: Running CMS in Subset Mode from DITTO

If you are running DITTO under CMS, you can press the PA2 key to enter CMS subset mode. In CMS subset mode, you can enter CMS commands while all action in your DITTO session is temporarily suspended. DITTO retains any values that are stored in a buffer for your session.

You can start another DITTO session from the CMS subset mode while your first session is suspended. However, you cannot go into the CMS subset from the second DITTO session. On leaving the second DITTO session, you return to the CMS subset, and can reenter your original DITTO session.

Switching between Full-Screen Mode and Line Mode

You can switch back and forth between full-screen mode and line mode. Simple operations may be quicker in line mode than in full-screen mode, whereas more complex operations are best suited to full-screen mode.

To switch from full-screen mode to line mode, enter **FSN** on the command line of a DITTO panel.

To switch from line mode back to full-screen mode, enter **FSY** in response to the DITTO function? prompt.

Handling Errors

If an error is detected in full-screen mode, DITTO issues an error message and the function ends. If the message has a number, you can get help information by pressing the Help function key.

If an error causes the system to cancel the task (for example, a file open error), DITTO tries to recover from the cancel condition and close any files left open. If recovery is not possible, or if you explicitly request cancel, DITTO is terminated.

Example of a Full-Screen Session

If you are a new DITTO user, you might want to try the following example full-screen session:

1. Enter DITTO from a TSO terminal (under MVS), a CICS terminal (under VSE), or a CMS terminal. The DITTO Task Selection menu appears.
2. Enter 13 on the command line to invoke the SET function. The SET entry panel appears.
3. Under MVS, set the PRINTOUT parameter to SYSOUT=c, where c is a SYSOUT class from A to Z or 0 to 9. Under VSE or CMS, set the PRINTOUT parameter to PRINTER.
4. Press the Exit function key to leave the SET function and return to the Task Selection menu.
5. Enter NEWS on the command line.
6. Use the Fwd and Bkwd function keys to move forward and backward through the News text.
7. Enter PRINT on the command line to print the current panel.
8. Press the Exit function key to exit the News function.
9. Enter PB on the command line.
10. Browse through the print output that you created (a screen of data from the News function). Press the Purge function key to delete the output.
11. If you want print output in future DITTO sessions to be sent to a different destination, invoke the SET function again and change the value of the PRINTOUT parameter.
12. Type E0J on the command line to end your DITTO session.

Running DITTO in Line Mode

In line mode, you can enter DITTO commands and parameters in a “question-and-answer” format. When you enter a DITTO command, DITTO prompts for more information as required.

Line mode is particularly useful if you are working at a system console where full-screen mode is not available.

Starting DITTO in Line Mode

To start DITTO in line mode, enter:

```
DITTO FSN
```

from a TSO terminal (under MVS), a CICS terminal (under VSE), or a CMS terminal.

To start DITTO at an MVS system console, enter `S DITTO` or start an equivalent started task as specified by the person who installed DITTO.

To start DITTO at a VSE system console, enter `EXEC DITTO`.

DITTO responds with this message:

```
*** DITTO/ESA active ***
```

```
When prompted, enter GO to skip optional prompts, or U to exit from function  
DITTO function?
```

Under MVS, the procedure for starting DITTO may depend on whether you are running DITTO APF-authorized. To find out which start procedure you should use, talk to your system administrator. For information about the implications of running DITTO APF-authorized, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

Under VSE, a GETVIS specification is *not* needed.

Entering Commands in Line Mode

In response to the `DITTO function?` prompt, enter a DITTO command. DITTO prompts you for the parameters needed. Commands and parameters are described in Part 3, “DITTO Reference” on page 147.

You can enter a command and its parameters on one line. Enter the parameters in the order shown in Part 3, “DITTO Reference” (under the heading “Command Syntax”), separated by commas or blanks. To accept default values, you can type a comma with no value.

For example, each of these line mode commands prints 5 files from the tape at address 181 (under VSE or CMS):

```
TP 181 CHAR U ALL 5  
TP 181,CHAR,U,ALL,5  
TP,181,,,5
```

To skip prompting for optional parameters, enter the required parameters followed by `GO`. For example, to print one file from the tape at address 181 you could enter any of the following:

```
TP 181 CHAR U ALL 1  
TP 181,CHAR,U,ALL,1  
TP 181 GO  
TP,181,GO
```

If DITTO cannot interpret a parameter, it prompts again for that parameter and for subsequent parameters.

Some parameters are not always prompted for. For example, SAM input parameters are not prompted for if a DLBL statement is provided. In cases where DITTO would not prompt for a parameter, DITTO does not look for the parameter in a single-line command.

VSE: Running DITTO in a VSE/ESA Dynamic Partition

You can run DITTO in a dynamic partition in line mode. Because of VSE/ESA restrictions for dynamic partitions, POWER spooling must be used for unit record device access.

Handling Errors

If an error is detected in line mode, DITTO issues an error message and the function ends.

If an error causes the system to cancel the task (for example, a file open error), DITTO tries to recover from the cancel condition and close any files left open. If recovery is not possible, or if you explicitly request cancel, DITTO is terminated.

To interrupt a DITTO function, enter U, END, or QUIT in response to a DITTO prompt. The DITTO function? prompt appears. You can now enter the name of a new DITTO function, or end the session by entering E0J.

Example of a Line Mode Session

If you are a new DITTO user, you might want to try the following example line mode session:

1. Enter DITTO FSN from a TSO terminal (under MVS), a CICS terminal (under VSE), or a CMS terminal.

2. To request a reference summary of DITTO functions, type:

```
XXX
```

The reference summary is printed. The print destination is determined by the current value of the PRINTOUT parameter of the SET function.

3. To end the DITTO session, type:

```
E0J
```

Executing a Single DITTO Function in Command Mode

In command mode, you enter a single command with all appropriate parameters from outside DITTO (for example, from an ISPF data set list, a CMS filelist, or a REXX procedure). After the command is executed, you automatically leave DITTO.

Under VSE, command mode is available *only* within a REXX procedure. Before the first DITTO command, specify:

```
CALL 'DITSETUP'  
ADDRESS DITTO
```

You can use either of the following syntaxes:

Parameter syntax

You enter DITTO followed by the command name and all appropriate parameters. Enter the parameters in the order shown in Part 3, "DITTO

Reference” (under the heading “Command Syntax”), separated by commas or blanks. To accept default values, you can type a comma with no value.

For example:

```
DITTO TP 181,,ALL,5
```

You must specify at least one parameter. If you enter only DITTO and the command name, you invoke DITTO in full-screen mode with the appropriate panel displayed.

Some parameters do not always apply. For example, SAM input parameters do not apply if a DLBL statement is provided. In cases where DITTO would not prompt for a parameter in line mode, DITTO does not look for the parameter in command mode.

If you omit an optional parameter, the default value is used. If you omit a required parameter, the command is not executed.

Command mode with the parameter syntax is very similar to line mode, as described in “Running DITTO in Line Mode” on page 19.

Keyword syntax

You enter DITTO followed by the command name prefixed by a dollar sign (\$). Enter the keywords shown in Part 3, “DITTO Reference.”

Each keyword and value must be separated by an equal sign or a blank. Keyword/value pairs are separated by a comma or a blank.

For example:

```
DITTO $TP INPUT=181,NLRECS=ALL,NFILES=5
```

or:

```
DITTO $TP INPUT 181 NLRECS ALL NFILES 5
```

If you omit an optional parameter, the default value is used. If you omit a required parameter, the command is not executed.

If you enter a list command (such as **SCS**) in command mode under MVS or CMS, full-screen mode DITTO is invoked and the appropriate list panel is displayed. Under VSE, the appropriate list is *printed* and you remain in line mode (or command mode).

Within a REXX procedure, you can use SET PRINTOUT=REXX to route print output to a REXX stem variable. For more information, see “DITTO Print Output” on page 26.

DITTO sets the following return codes:

- 0** All DITTO functions processed successfully.
- 16** DITTO terminated because of an error. DITTO or system messages identify the error.
- >100** Not enough storage to load DITTO.

Additionally, some functions have function-specific return codes. If you receive a return code that is not listed above, refer to the section describing the function in Part 3, “DITTO Reference.”

Deciding Which Syntax to Use

The parameter syntax is most useful in the following cases:

- Simple commands with very few parameters, such as DITTO REW 181
- In a selection list (under MVS or CMS), where you want to use as few keystrokes as possible
- In existing procedures that were written before the keyword syntax became available.

The keyword syntax is most useful in the following cases:

- Complicated commands with many parameters, where it may be difficult to figure out the correct parameter sequence
- In procedures, where documentation and maintenance are concerns. This command:

```
DITTO $TP INPUT=181,NLRECS=ALL,NFILES=5
```

is much easier to understand than this command:

```
DITTO TP 181,,ALL,5
```

Example of a Command Mode Procedure

Figure 22 shows a REXX procedure that invokes DITTO in command mode. The CALL 'DITSETUP' and ADDRESS DITTO statements apply in VSE only.

```

/* REXX */
/* *****/
/* Example: Execute DITTO commands from a REXX procedure */
/* *****/
parse upper source environment.
if environment='VSE' then do
  CALL 'DITSETUP'          /* setup DITTOs REXX interface */
  ADDRESS DITTO           /* pass commands to DITTO */
end
/* List the catalog entries on the printout */
DITTO '$SCS'              /* call SCS for the master catalog */
say 'SCS return code:' RC
/* Route subsequent print output to DITTO.n REXX variable */
DITTO '$SET PRINTLEN=80,PRINTOUT=REXX' /* call the SET command */
say 'SET return code:' RC
ditto.0=0                 /* discard any previous output */
                          /* call SCS for a user catalog */
DITTO '$SCS CATALOG=VSESP.USER.CATALOG,DSNAME=VSE.** ,SORTBY=DATE'
say 'SCS return code:' RC
/* Display the DITTO.n variable contents */
SAY '>>> SCS output:'
DO i=1 to ditto.0        /* process all printed lines */
  SAY ditto.i           /* display the printline */
END

```

Figure 22. Command Mode Example (REXX Procedure)

Running DITTO in Batch Mode

DITTO functions that do not require user interaction are available in batch mode. Batch mode is most useful for:

- Routine jobs that you want to run regularly
- Jobs that combine DITTO functions
- Jobs that produce large amounts of printed output
- Jobs that require longer processing times, such as copying large files.

DITTO sets the following return codes:

- 0** All DITTO functions processed successfully.
- 16** DITTO terminated because of an error. DITTO or system messages identify the error.
- >100** Not enough storage to load DITTO.

Additionally, some functions have function-specific return codes. If you receive a return code that is not listed above, refer to the section describing the function in Part 3, "DITTO Reference."

This book assumes that you know how to code job control statements.

MVS: Running DITTO in Batch Mode

To use DITTO in a batch job under MVS, include the following EXEC statement:

```
// EXEC PGM=DITTO
```

You must supply a DD statement for SYSPRINT. You might also need a STEPLIB statement or other DD statements, depending on which DITTO commands the batch job contains.

VSE: Running DITTO in Batch Mode

To indicate that DITTO control statements are to be processed in a batch job, include the following information in your job control:

```
// UPSI 1                               Batch Operation  
   <input and output assignments>  
// EXEC DITTO
```

For file functions, you can supply TLBL statements for tape files, and DLBL and EXTENT statements for disk files. You can supply these statements either within the job stream or by cataloging them into the partition or system standard label area. You then specify the DLBL or TLBL file name in the FILEIN or FILEOUT parameter or specify the programmer logical unit (SYS nnn) in the INPUT or OUTPUT parameter.

A GETVIS specification is *not* needed.

You can run DITTO in a dynamic partition in batch mode. Because of VSE/ESA restrictions for dynamic partitions, POWER spooling must be used for unit record device access.

CMS: Running DITTO in Batch Mode

To start a batch job under CMS, enter:

```
DITTO filename [filetype|DITTO [filemode]]
```

where *filename*, *filetype*, and *filemode* are the filename, filetype, and filemode of a file that contains DITTO control statements.

Coding DITTO Control Statements

The rules for coding DITTO control statements are:

- Each control statement begins with \$\$DITTO in columns 1 to 7, followed by a blank in column 8, followed by the command name, followed by the first keyword and parameter (if any).
- Parameters must be specified with a keyword and an equal sign, like this:
KEYWORD=*parameter*.
- Each parameter, except the last parameter, is followed immediately by a comma.
- To continue a command onto the next line, you do not need a continuation character; simply end the first line with a comma and begin the next line with \$\$DITTO and a blank.

For example, this batch command prints 5 files from the tape at address 181 (under VSE or CMS):

```
$$DITTO TP INPUT=181,NBLKS=ALL,NFILES=5
```

This is the same command, split over three lines:

```
$$DITTO TP INPUT=181,  
$$DITTO     NBLKS=ALL,  
$$DITTO     NFILES=5
```

Commands and parameters are described in Part 3, "DITTO Reference" on page 147.

Handling Errors

If an error is detected in batch mode, DITTO issues an error message and terminates. If an error causes the system to cancel the job (for example, a file open error), DITTO tries to recover from the cancel condition and close any files left open before it terminates.

Examples of a Batch Job

If you want to code a practice DITTO batch job, use the job shown in Figure 23 on page 26, Figure 24 on page 26, or Figure 25 on page 26 as a model. These jobs list a summary of DITTO functions.

```
//DITTOJOB (Job Information)
//DITTO EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$DITTO XXX
$$DITTO EOJ
/*
//
```

Figure 23. Batch Job Example: Listing DITTO Functions (MVS)

```
// JOB XXX PRINT LIST OF DITTO COMMANDS
// UPSI 1
// EXEC DITTO
$$DITTO XXX
$$DITTO EOJ
/&
```

Figure 24. Batch Job Example: Listing DITTO Functions (VSE)

```
$$DITTO XXX
$$DITTO EOJ
```

Figure 25. Batch Job Example: Listing DITTO Functions (CMS)

There are more examples of JCL and DITTO commands in Part 2, “Guide to Using DITTO Functions” and Part 3, “DITTO Reference.”

DITTO Print Output

Any print output that you produce during a DITTO session is handled as follows:

- In batch mode, print output is always sent to `SYSPRINT` (MVS), `SYSLST` (VSE), or the virtual printer (CMS)
- In line mode from the VSE system console, print output is always sent to `SYSLST`.
- In any other mode the destination for print output depends on the value of the `PRINTOUT` keyword in the **SET** function:

SYSPRINT MVS only: Output is sent to the current `SYSPRINT` allocation.

SYSOUT=c MVS only: Output is accumulated in a temporary print file. When you end your DITTO session, the output is sent to `SYSOUT` class `c`.

You can use the **PB** (Print Browse) function to browse the temporary print file during a full-screen DITTO session. When browsing, you can enter the `PURGE` command to purge the print output. When you leave the `PB` function, output stops accumulating. Any subsequent print output creates a new temporary file and sends the previous file to `SYSOUT` class `c`.

- TERMINAL** Output is sent to the terminal.
- REXX** Output is written to a REXX stem variable. Each line of output corresponds to a variable named DITTO.*nnn*. DITTO.0 contains a line count.
- After your first printing operation, lines of output are written in variables DITTO.1 to DITTO.*n*, and DITTO.0 has the value *n*. The next printing operation will begin with DITTO.*n+1*.
- PRINTER** VSE or CMS only: Output is accumulated in a spool file. When you end your DITTO session, the spool is closed.
- You can use the **PB** (Print Browse) function to browse the spool file during a full-screen DITTO session. When you enter the PB function, the spool file is closed. When browsing, you can enter the PURGE command to purge the print output. For any subsequent print output, a new spool file is created.

VSE: System File Assignments

SYSLST can be assigned to printer, tape, or direct access storage devices (DASD). If you assign SYSLST to a DASD device, the lines are truncated to 121 characters. System file assignments to DASD also require appropriate DLBL and EXTENT statement information. (For more information, see *VSE/ESA Guide to System Functions*.)

SYSIPT and SYSPCH can be assigned to card, tape, or DASD devices.

When DITTO runs in a procedure with DATA=YES, a label for IJSYSIN must be present in the label area. A dummy label is sufficient; it may be specified as:

```
// DLBL IJSYSIN, 'DUMMY'
// EXTENT SYSIN
```

VSE: Functions Supported in Stand-Alone Environment

A subset of DITTO functions is available in the VSE stand-alone environment. You invoke stand-alone DITTO when you IPL your system from a VSE stand-alone system backup tape.

The following tape functions are supported:

| | | | | |
|-----|-----|-----|-----|-----|
| BSF | FSR | TCN | TP | TTC |
| BSR | INT | TLB | TRL | TTR |
| ERT | REW | TLT | TRS | WTM |
| FSF | RUN | TMP | TT | |

The following disk functions are supported:

| | | | | |
|-----|-----|-----|-----|-----|
| DCN | DP | DRS | DVT | PVT |
| DID | DRL | DSE | EOF | |

The following miscellaneous functions are supported:

| | | | | |
|-----|-----|-----|-----|-----|
| EOJ | LVL | SET | VER | XXX |
|-----|-----|-----|-----|-----|

Modes

Part 2. Guide to Using DITTO Functions

For users new to DITTO

Part 1, "Introduction to DITTO" on page 1 gives an overview of DITTO functions, and the different ways you can use DITTO.

For users already familiar with DITTO basics

Part 2, "Guide to Using DITTO Functions" describes what you can accomplish with DITTO functions. The first two chapters describe browse and edit functions that are available in full-screen mode. The remaining chapters are divided according to the type of input you are working with: catalog entries, tapes, disk records, sequential data (QSAM or SAM), VSAM data, CMS files, VSE library members, OAM objects, or cards.

For experienced DITTO users

Part 3, "DITTO Reference" on page 147 presents the DITTO functions in alphabetical order.

For special situations

Part 4, "Appendixes" on page 347 contains information on compatibility with previous releases, installing DITTO, controlling DITTO security, and responding to DITTO error messages.

If you are running DITTO in full-screen mode, you can also refer to the online help.

Chapter 3. Browsing Data in Full-Screen Mode

In full-screen mode, you can use the following browse functions:

Figure 26. Full-Screen Browse Functions

| Purpose | Functions | |
|--|------------|------------------------|
| Browse physical tape records | TB | Tape Browse |
| Browse physical disk records | DB | Disk Browse |
| Browse CMS files | FB | CMS File Browse |
| Browse CP spool files or VSE/POWER spool files | SPB | Spool Browse |
| Browse VSAM data sets | VB | VSAM Browse |
| | DVB | Data in Virtual Browse |
| Browse library members | LB | Library Member Browse |
| Browse OAM objects | OB | Object Browse |
| Browse user storage | MB | Memory Browse |
| Browse print output | APB | Advanced Print Browse |
| | PB | Print Browse |

Browsing Data

In full-screen mode, you can use DITTO's browse functions to look at data without changing it. There are three ways to select a browse function:

- Select the function from a full-screen menu.
- Enter a browse command:
 - TB** To browse physical records in a tape file
 - DB** To browse physical records on a disk
 - VB** To browse logical records in a VSAM data set
 - DVB** MVS: to browse a VSAM data-in-virtual data set
 - FB** CMS: to browse a CMS file
 - SPB** VSE or CMS: to browse a VSE/POWER list queue or CP spool (reader queue) file
 - LB** VSE: to browse a library member
 - OB** MVS: to browse an OAM object
 - MB** To browse user storage
 - PB** To browse print output, as described in "DITTO Print Output" on page 26
 - APB** To browse Advanced Function Printing (AFP) format data (such as LIST3820 files).
- Enter a list command:
 - DVT** To list the files on a disk (by displaying the disk VTOC)
 - VDL** To list VSAM data sets
 - LDL** VSE: to list library members
 - DLA** VSE: to display the contents of the label area
 - SPL** VSE: to list entries in the POWER LST queue
 - SPS** VSE: to list (and issue commands on) entries in any POWER queue (LST, PUN, RDR, or XMT)
 - ODL** MVS: to list OAM objects
 - SCS** VSE or MVS: to list cataloged data sets

On the list panel that appears, type browse (view for APB) over the *beginning* of the name of the file that you want to browse, and press Enter. (You can also select the Browse or View function from the Process pull-down menu in a list panel.)

If necessary, DITTO prompts you for the name of the file or object that you want to browse. The **VB** (VSAM Browse) function also asks if you want to view records or control intervals. (In this chapter, all references to records also apply to VSAM control intervals.)

There are three types of browse functions:

- The **TB**, **DB**, **VB**, **FB**, **SPB**, **APB**, and **PB** functions display a collection of *records*.
- The **DVB**, **MB**, and **OB** functions display a collection of *bytes*.
- The **LB** function displays a collection of bytes if the library member is stored in string format; otherwise, it displays a collection of records.

The browse functions are described in more detail in the sections that follow.

Browsing a Collection of Records

The **TB**, **DB**, **VB**, **FB**, **SPB**, **APB**, and **PB** functions (and the **LB** function when the library member is not in string format) use similar panels, which display several records at a time. Figure 27 shows the VSAM Browse panel, which is displayed when you enter the **VB** command.

```

Process   View   Options   Help
-----
DITTO/ESA for VSE          VB - VSAM Browse

RBA 0
VOLSER SYSWK2   Type ESDS   DSNAME DTT.ITT.ESDS           Col 1           Format CHAR

RBA      LEN      1...5...10...5...20...5...30...5...40...5...50...5...60
0         80      1 * F * a * This is the 1st record of DTT.ITT.ESDS *****
80        120     2 * E * d * This is the 2nd record of DTT.ITT.ESDS *****
200       60      3 * D * b * This is the 3rd record of DTT.ITT.ESDS *****
260      100     4 * C * e * This is the 4th record of DTT.ITT.ESDS *****
360       80      5 * B * c * This is the 5th record of DTT.ITT.ESDS *****
440       80      6 * A * f * This is the 6th record of DTT.ITT.ESDS *****
**** End of data ****

Command ==> _____ Scroll PAGE
F1=Help F2=Zoom F3=Exit F4=Left F5=Right F6=Rfind F7=Bkwd F8=Fwd
F10=Actions F11=CRetrieve F12=Cancel

```

Figure 27. VSAM Browse Panel

Changing the Data Display Format

To change the data display format, enter one of the following values in the Format field:

- CHAR** For character format (this is the default)
- HEX** For hexadecimal across format (system dump format)
- UPDN** For hexadecimal updown format (DITTO dump format)

For an example of each format, see “Formats for Displaying or Printing Data” on page 4.

Locating a Specific Record

You can locate a specific record in the following ways:

- Scroll the data:
 1. In the Scroll field, type one of the following:
 - PAGE** To scroll one page of data at a time
 - HALF** To scroll half a page of data at a time
 - DATA** To scroll one line less than a page of data at a time
 - MAX** To scroll to the top or bottom of the file (or beginning or end, when scrolling left and right)
 - CSR** To scroll to the cursor position
 - nnnn** To scroll *nnnn* lines or columns at a time

2. Use the Fwd and Bkwd function keys (or the DOWN and UP commands) to scroll forward and backward by the amount indicated in the Scroll field.

- Locate a line number. If the CMDSYNTAX=ISPF processing option is in effect (this is the default under MVS), then in some functions you can type the following and press Enter to move to a particular line number:

```
LOCATE line_number
```

For a list of the functions in which you can use this form of LOCATE, and more information about the CMDSYNTAX processing option, see “SET (Set Processing Options)” on page 242.

- Find a string, by typing one of the following and pressing the Enter key:

```
FIND string  
/ string
```

where *string* contains up to 50 characters. The string can be:

- A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in single quotation marks. The string *can* contain blanks and commas; the string must match exactly (including case).
- X followed by a hexadecimal string enclosed in single quotation marks, for example X'C1C2C3'.

If the CMDSYNTAX=CMS (default under VM) or CMDSYNTAX=DITTO (default under VSE) processing option is in effect, then you can use LOCATE as a synonym for FIND:

```
LOCATE string
```

The commands FIND and LOCATE may be abbreviated.

The first record (after the current position) that contains the string is moved to the top position. To find the next occurrence of the string, press the Rfind function key.

- Use other input fields:
 - In the **DB** function, you can enter a disk location in the CYL-HD-REC field.
 - In the **TB** function, you can enter a block number in the Blk field. You can also enter a tape-positioning command on the command line. For example, you can use the **FSF** or **BSF** commands to view a different tape file.
 - In the **VB** function, you can:
 - Enter a relative byte address in the RBA field
 - Enter a key value in the Key field (for a key-sequenced data set)
 - Enter a slot value in the Slot field (for a relative record data set)

Note: If you use the Key field, the records are sorted by key. To display the records in ascending order by byte address, use the RBA field. For VSAM files that do not allow positioning by byte address, DITTO prevents you from entering a value in the RBA field.

Seeing More of a Record

There are three ways to see more of a record:

- Use the Right or Left function key (or command) to scroll to the right or left.
- Enter a value in the Col field to see records starting at the specified column.
- To see a detailed view of a single record, select the record (by moving the record to the top of the screen or putting the cursor on the record), and press the Zoom function key or enter the ZOOM command.

This takes you into single-record mode. Figure 28 shows the VSAM Browse panel in single-record mode.

```

Process   View   Options   Help
-----
DITTO/ESA for VSE           VB - VSAM Browse

RBA 5137          Key 'EAHIESE0303 '          Col 1          Format CHAR
VOLSER DOSRES   Type KSDS   DSNAME TEXT.REPSTORY.FILE

LEN          BYTE 1...5...10...5...20...5...30...5...40...5...50...5...60
500          1 EAHIESE0303 ...THE SPECIFIED PARTITION ID IS INVALID OR PAR
              61 TITION IS NOT ALLOCATED....You entered a partition ID that
              121 is unknown to the system...Either the partition ID is not va
              181 lid or the partition is..not allocated....If the partition
              241 ID starts with letter 'B' or 'F' a static..partition is assu
              301 med...Valid partition IDs are 'BG' or 'F1', 'F2', ..., 'FB'.
              361 ...Otherwise, a dynamic partition ID is assumed.. Valid par
              421 tition IDs are listed in the panel 'Dynamic Partitions'..for
              481 the class selected.

Command ==>
F1=Help F2=Zoom F3=Exit F4=Left F5=Right F6=Rfind F7=Bkwd F8=Fwd
F10=Actions F11=CRetrieve F12=Cancel
  
```

Figure 28. VSAM Browse Single-Record Panel

To see a different record, enter a value in an input field. For example, if you are viewing a record of a VSAM key-sequenced data set, you can enter a value in either the RBA field or the Key field.

To return to multiple-record mode, press the Zoom function key again.

Saving Data in a Buffer

Enter `PUT col1-col2` to save the data from column `col1` to column `col2` (of the record at the top of the panel) in a buffer. If you enter only one column number, the single character in that column is saved in a buffer. You can later retrieve this data by using the `GET` command.

For example, while browsing a tape, you can use `PUT` to copy a record into a buffer. Later, while updating a VSAM file, you can use `GET` to get the data and add it to the file.

To see the data stored in the buffer, use the command `MB CLIPBOARD`. The length of the buffer data is stored beginning with offset `X'C'`; the data itself is stored beginning with offset `X'10'`.

Browse

Printing Data

To print some of the data that you are browsing, enter one of the following commands:

- PRINT** To print the current screen image
- RD** To print the selected record, in hexadecimal format
- RP** To print the selected record, in character format

Purging Print Output and Spool Files

If you are browsing print output (**PB**) or a spool file (**SPB**), you can press the Purge function key (or enter PURGE on the command line) to purge the file.

Leaving the Browse Function

Press the Cancel or Exit function key (or enter EXIT on the command line) to leave the browse function.

Browsing a Collection of Bytes

The **DVB**, **MB**, and **OB** functions, and the **LB** function when the library member is in string format, use similar panels, which display a collection of bytes. The **MB** function displays data only in hexadecimal across format.

Figure 29 shows an example of the Memory Browse panel.

```
Process  View  Options  Help
-----
DITTO/ESA for VSE          MB - Memory Browse

Browse address 00000000          Search limit * _____

Location  Offset  -----Hex-----  -----Char-----
00000000      0  00000000  00D300D3  00000000  00000000  *.....L.L.....*
00000010     10  00295000  002CE398  07DD0000  000CBD3E  *..&...Tq.....*
00000020     20  07DD0000  00553F86  07DD2000  000AE9E6  *.....f.....ZW*
00000030     30  00000000  00000000  070F0000  00002FD8  *.....Q*
00000040     40  0007FD88  0C000000  0007FD60  00000000  *..h.....-...*
00000050     50  FF000000  00000000  040C0000  0000B804  *.....*
00000060     60  040C0000  0000B71E  000C0000  0000B9FC  *.....*
00000070     70  04080000  0001400E  040C0000  0000B872  *.....*
00000080     80  00000410  00001000  0002003E  00040040  *.....*
00000090     90  017C3000  00040000  00000000  0000000A  *..@.....*
000000A0    A0  00000000  00000000  00000000  00000000  *.....*
000000B0    B0  00000000  00000000  00000244  00003098  *.....q*
000000C0    C0  00000000  00000000  00000000  00000000  *.....*
```

Command ==> _____ Scroll PAGE
F1=Help F3=Exit F4=Previous F5=Next F6=Rfind F7=Bkwd F8=Fwd F10=Actions
F11=CRetrieve F12=Cancel

Figure 29. Memory Browse Panel

Changing the Data Display Format

To change the data display format within **DVB**, **LB**, or **OB**, enter one of the following values in the Format field:

- CHAR** For character format (this is the default)
- HEX** For hexadecimal across format (system dump format)
- UPDN** For hexadecimal updown format (DITTO dump format)

For an example of each format, see “Formats for Displaying or Printing Data” on page 4.

Locating Specific Data

You can locate specific data in the following ways:

- Scroll the data:
 1. In the Scroll field, type one of the following:
 - PAGE** To scroll one page of data at a time
 - HALF** To scroll half a page of data at a time
 - DATA** To scroll one line less than a page of data at a time
 - MAX** To scroll to the top or bottom of the file (or beginning or end, when scrolling left and right)
 - CSR** To scroll to the cursor position
 - nnnn** To scroll *nnnn* lines or columns at a time
 2. Use the Fwd and Bkwd function keys (or the DOWN and UP commands) to scroll forward and backward by the amount indicated in the Scroll field.

- Locate a line number. If the CMDSYNTAX=ISPF processing option is in effect (this is the default under MVS), then in some functions you can type the following and press Enter to move to a particular line number:

```
LOCATE line_number
```

For a list of the functions in which you can use this form of LOCATE, and more information about the CMDSYNTAX processing option, see “SET (Set Processing Options)” on page 242.

- Find a string, by typing one of the following and pressing the Enter key:

```
FIND string
/string
```

where *string* contains up to 50 characters. The string can be:

- A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in single quotation marks. The string *can* contain blanks and commas; the string must match exactly (including case).
- X followed by a hexadecimal string enclosed in single quotation marks, for example X'C1C2C3'.

If the CMDSYNTAX=CMS (default under VM) or CMDSYNTAX=DITTO (default under VSE) processing option is in effect, then you can use LOCATE as a synonym for FIND:

```
LOCATE string
```

The commands FIND and LOCATE may be abbreviated.

Browse

The first occurrence of the string (after the current position) is moved to the top position. To find the next occurrence of the string, press the `Rfind` function key.

To limit the search to a specified number of bytes (in **DVB** or **MB**), type a hexadecimal number in the `Search Limit` field.

- Use other input fields. You can:
 - Enter an address in the field labeled `Browse address` or `Offset`
 - Enter a number with a plus sign (+) or minus sign (–) on the command line, to move forward or backward the specified number of bytes.
 - Use the `Next` and `Previous` function keys (in **DVB** or **MB**):
 - Move the cursor to a position on the panel and press the `Next` function key, to make the selected address (or offset) the current address. DITTO saves up to 64 addresses that you select in this way.
 - With the cursor on the command line, press the `Previous` function key or the `Next` function key, to move backwards or forwards through the list of saved addresses.

Saving Data in a Buffer

In the **DVB**, **LB**, or **OB** function, enter `PUT start-end` to save the data from byte *start* to byte *end* (relative to the current browse address) in a buffer. If you enter only one number, the single character in that position is saved in a buffer. You can later retrieve this data by using the `GET` command.

For example, while browsing a library member in string format, you can use `PUT` to copy several bytes into a buffer. Later, while updating a VSAM file, you can use `GET` to get the data and add it to the file.

To see the data stored in the buffer, use the command `MB CLIPBOARD`. The length of the buffer data is stored beginning with offset `X'C'`; the data itself is stored beginning with offset `X'10'`.

Printing Data

At any time, you can enter `PRINT` on the command line to print the current screen image.

Leaving the Browse Function

Press the `Cancel` or `Exit` function key (or enter `EXIT` on the command line) to leave the browse function.

Chapter 4. Editing Data in Full-Screen Mode

In full-screen mode, you can use the following edit functions:

Figure 30. Full-Screen Edit Functions

| Purpose | Functions | |
|----------------------|------------------|---------------------|
| Edit library members | LE | Library Member Edit |
| Edit VSAM data sets | VE | VSAM Edit |
| Edit disk tracks | DTE | Disk Track Edit |

Editing Data

In full-screen mode, you can use DITTO's edit functions to look at data and change it as required. There are three ways to select an edit function:

- Select the function from a full-screen menu.
- Enter an edit command:
 - LE** VSE: to edit fixed-format library members; for string-format members, **LU** is used
 - VE** To edit data in a VSAM data set
 - DTE** To edit a disk track
- Enter a list command:
 - DVT** To list the files on a disk
 - VDL** To list VSAM data sets
 - LDL** VSE: to list library members
 - LSL** VSE: to list library status information
 - DLA** VSE: to display the contents of the label area
 - SCS** VSE or MVS: to list cataloged data sets

On the list panel that appears, type `edit` over the *beginning* of the name of the data set or library member that you want to edit, and press Enter. (You can also select the Edit function from the Process pull-down menu in a list panel.)

Figure 31 shows the VSAM Edit panel, which is displayed when you enter the **VE** command. (A similar panel is displayed for **DTE** or **LE**.)

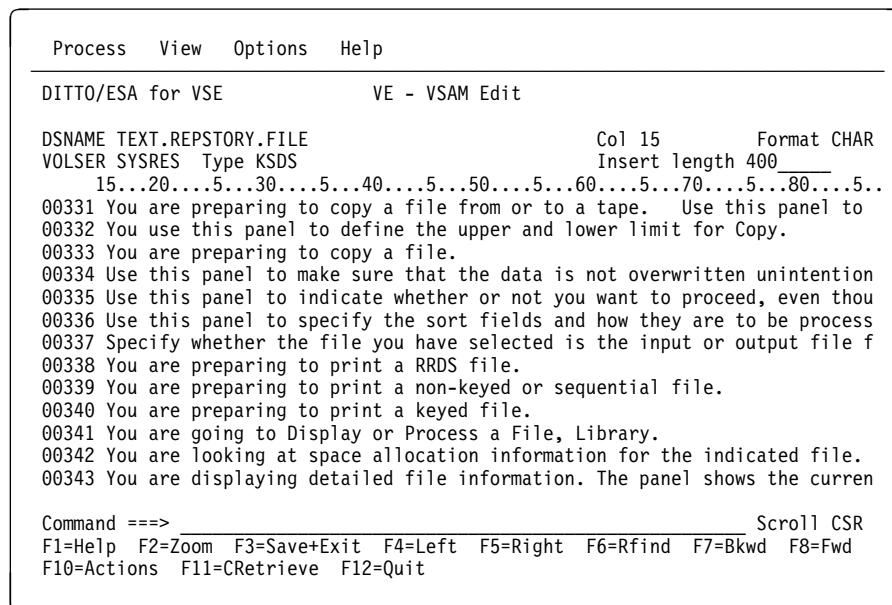


Figure 31. VSAM Edit Panel

Figure 31 shows a panel with a prefix area on the left of the screen. You can use this prefix area to enter various commands. You can also enter commands on the command line and change the values of input fields. These possibilities are described in more detail in the remainder of this chapter.

Changing the Data Display Format

To change the data display format, enter one of the following values in the Format field:

- CHAR** For character format (this is the default)
- HEX** For hexadecimal across format (system dump format)
- UPDN** For hexadecimal updown format (DITTO dump format)

For an example of each format, see “Formats for Displaying or Printing Data” on page 4.

Locating a Specific Record

You can locate a specific record in the following ways:

- Scroll the data:
 1. In the Scroll field, type one of the following:
 - PAGE** To scroll one page of data at a time
 - HALF** To scroll half a page of data at a time
 - DATA** To scroll one line less than a page of data at a time
 - MAX** To scroll to the top or bottom of the file (or beginning or end, when scrolling left and right)
 - CSR** To scroll to the cursor position
 - nnnn** To scroll *nnnn* lines or columns at a time
 2. Use the Fwd and Bkwd function keys (or the DOWN and UP commands) to scroll forward and backward by the amount indicated in the Scroll field.

- Locate a line number. If the CMDSYNTAX=ISPF processing option is in effect (this is the default under MVS), then in some functions you can type the following and press Enter to move to a particular line number:

```
LOCATE line_number
```

For a list of the functions in which you can use this form of LOCATE, and more information about the CMDSYNTAX processing option, see “SET (Set Processing Options)” on page 242.

- Find a string, by typing one of the following and pressing the Enter key:

```
FIND string
/string
```

where *string* contains up to 50 characters. The string can be:

- A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in single quotation marks. The string *can* contain blanks and commas; the string must match exactly (including case).
- X followed by a hexadecimal string enclosed in single quotation marks, for example X'C1C2C3'.

If the CMDSYNTAX=CMS (default under VM) or CMDSYNTAX=DITTO (default under VSE) processing option is in effect, then you can use LOCATE as a synonym for FIND:

```
LOCATE string
```

The commands FIND and LOCATE may be abbreviated.

If the first record after the current position that contains the string is not visible on the screen, it is moved to the top position. To find the next occurrence of the string, press the Rfind function key.

Seeing More of a Record

There are three ways to see more of a record:

- Use the Right or Left function key (or command) to scroll to the right or left.
- Enter a value in the Col1 field to see records starting at the specified column.
- To see a detailed view of a single record, select the record (by moving the record to the top of the screen or putting the cursor on the record), and press the Zoom function key or enter the ZOOM command.

This takes you into single-record mode. Figure 32 shows the VSAM Edit panel in single-record mode.

```

Process  View  Options  Help
-----
DITTO/ESA for VSE          VE - VSAM Edit

DSNAME TEXT.REPSTORY.FILE          Col 1          Format CHAR
VOLSER SYSRES  Type KSDS          Insert length 400
Record Length Byte 1...5...10...5...20...5...30...5...40...5...50...5...60
538   388   1  EATDUMME1  ..This panel is used to enter messages for other
        61  users...Key your message into the area where the cursor..is
        121 located....Press ENTER to have the message formatted and d
        181 isplayed..to you.  You can then change the text, if you wish
        241 ....Use the Message Complete key when you are finished..ent
        301 ering the message....Use the Cancel Message key if you deci
        361 de not to send..the message.

Command ==> _____ Scroll CSR
F1=Help  F2=Zoom  F3=Save+Exit  F4=Left  F5=Right  F6=Rfind  F7=Bkwd  F8=Fwd
F10=Actions  F11=CRetrieve  F12=Quit
    
```

Figure 32. VSAM Edit Single-Record Panel

To return to multiple-record mode, press the Zoom function key again.

Changing the Length of a Record (DTE and VE Only)

For a disk track or any VSAM data set except an ESDS or RRDS, the length of a record can be changed. To change the length of a record, use the Zoom function to display the record you require in single-record mode, and enter the required record length in the Length field.

In DTE, specify a length of 0 to create an EOF record.

Setting the Length for Inserted Records (DTE and VE Only)

Use the Insert Length field to set up the length for new (inserted) records. The insert length must not exceed the maximum record length and must be large enough to hold any key field. For RRDS data sets, this field is protected.

Adding a Key Field to a Disk Record (DTE Only)

To add a key field to a disk record, use the Zoom function to display the record you require in single-record mode, and enter the following command:

```
ADDKEY [length]
```

This command adds a key with the specified length (default 44, maximum 255) to the disk record.

Mapping the Track Contents (DTE Only)

Use the MAP command to display a summary of the track usage and free space.

Locating and Changing a String

Use the CHANGE command from any edit panel to locate and replace one or more occurrences of a string.

Note: In single-record mode, the CHANGE command affects only the current record.

The syntax of the CHANGE command depends on the value of the CMDSYNTAX processing option:

CMDSYNTAX=CMS

The CHANGE command has syntax similar to the CMS editor CHANGE command:

CHANGE */old/new/* Changes the first occurrence in the current record only.

CHANGE */old/new/ n* In multiple-record mode: changes the first occurrence in the current record, and the first occurrence in each of the following *n-1* records.

In single-record mode: changes the first occurrence in the current record only.

CHANGE */old/new/ ** In multiple-record mode: changes the first occurrence in the current record, and the first occurrence in each subsequent record.

In single-record mode: changes the first occurrence in the current record only.

CHANGE */old/new/ ALL* Same as above.

| | | |
|--|--------------------------|--|
| | CHANGE /old/new/ * n | In multiple-record mode: changes the first <i>n</i> occurrences in the current record, and the first <i>n</i> occurrences in each subsequent record. |
| | | In single-record mode: changes the first <i>n</i> occurrences in the current record only. |
| | CHANGE /old/new/ 1 * | Changes all occurrences in the current record. |
| | CHANGE /old/new/ * * | In multiple-record mode: changes all occurrences in the current record and in all subsequent records. |
| | | In single-record mode: changes all occurrences in the current record only. |
| | CHANGE /old/new/ ALL ALL | Same as above. |

| The first non-blank delimiter is used as a delimiter (in the examples above, the / character). If *old* is a null string, the *new* string is inserted at the current column position.

| The *old* string is case sensitive. For example:

| CHANGE /before/after/
 | changes only "before," not "BEFORE" or "Before."

| If CAPS ON or CASE UPPER is in effect, the CHANGE command behaves as if the *new* string was specified in all uppercase. For details, see "Translating Input to Uppercase" on page 46.

| This is the default under VM.

| **CMDSYNTAX=DITTO**

| The CHANGE command has the same syntax as in DITTO/ESA Release 2 and earlier:

| | | |
|--|----------------------|--|
| | CHANGE /old/new/ | In multiple-record mode: changes the next occurrence. |
| | | In single-record mode: changes the next occurrence in the current record only. |
| | CHANGE /old/new/ n | In multiple-record mode: changes the next <i>n</i> occurrences. |
| | | In single-record mode: changes the next <i>n</i> occurrences in the current record only. |
| | CHANGE /old/new/ * | In multiple-record mode: changes the next occurrence and all subsequent occurrences. |
| | | In single-record mode: changes all occurrences in the current record. |
| | CHANGE /old/new/ ALL | Same as above. |

| The first non-blank delimiter is used as a delimiter (in the examples above, the / character). If *old* is a null string, the *new* string is inserted at the current column position.

The *old* string is case sensitive.

If CAPS ON or CASE UPPER is in effect, the CHANGE command behaves as if the *new* string was specified in all uppercase. For details, see “Translating Input to Uppercase” on page 46.

This is the default under VSE.

CMDSYNTAX=ISPF

The CHANGE command has syntax similar to the ISPF editor CHANGE command:

CHANGE *old new* In multiple-record mode: Changes the next occurrence.

In single-record mode: Changes the next occurrence in the current record only.

CHANGE *old new ** In multiple-record mode: changes all occurrences in all records.

In single-record mode: changes all occurrences in the current record.

CHANGE *old new ALL* Same as above.

The *old* string can be:

- A character string without blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in single quotation marks. It can contain blanks and commas. The string must match exactly (including case).
- X followed by a hexadecimal string enclosed in single quotation marks. (for example, X'C1C2').

The *new* string can be:

- A character string without blanks or commas. Case is respected and retained, unless CAPS ON or CASE UPPER is in effect, in which case the CHANGE command behaves as if the *new* string was specified in all uppercase. For details, see “Translating Input to Uppercase” on page 46.
- A character string enclosed in single quotation marks. It can contain blanks and commas. Case is respected and retained, unless CASE UPPER is in effect, in which case the CHANGE command behaves as if the *new* string was specified in all uppercase. For details, see “Translating Input to Uppercase” on page 46.
- X followed by a hexadecimal string enclosed in single quotation marks. (for example, X'C1C2'). If CASE UPPER is in effect, the CHANGE command behaves as if the hexadecimal string for the equivalent uppercase characters was specified. For details, see “Translating Input to Uppercase” on page 46.

If *old* is a null string (' '), the *new* string is inserted at the current column position.

This is the default under MVS.

For more information about the CMDSYNTAX processing option, see “SET (Set Processing Options)” on page 242.

After entering a CHANGE command, you can use the RChange function key to change the next occurrence of the string. To locate the next occurrence of the string, and optionally change it, use a combination of the RFind and RChange function keys.

When using the CHANGE command to change data in an ESDS, RRDS, or VRDS VSAM data set, the lengths of the old string and the new string in the CHANGE command must be the same. The CHANGE command cannot be used to change the record length of a record in an ESDS, RRDS, or VRDS data set.

If the changed line is too long for the file, it is truncated.

Translating Input to Uppercase

The CAPS and CASE commands control whether data in the data area is translated into uppercase on input (CAPS, CAPS ON, or CASE UPPER), or left alone (CASE, CASE MIXED, or CAPS OFF; this is the default).

The CAPS ON and CASE UPPER commands have slightly different effects:

CAPS or CAPS ON

Is similar to the ISPF editor CAPS command:

- If you overwrite any data in a record, the entire record is translated to uppercase.
- The CHANGE command treats the *new* (replacement) string as if had been specified in all uppercase. There are two exceptions: if SET CMDSYNTAX=ISPF is in effect, and the *new* string is enclosed in single quotation marks or specified as a hexadecimal string, then the *new* string is not translated to uppercase.

CAPS or CAPS ON only affects records changed after the CAPS or CAPS ON command is entered.

CASE UPPER

Translates to uppercase all data in any new or changed records, regardless of how the records are changed. CASE UPPER affects all records inserted or changed in the current edit session, including records inserted or changed before the CASE UPPER command is entered.

The CAPS OFF, CASE, and CASE MIXED commands have the same effect: they turn off translation to uppercase.

Any CAPS command overrides any previously entered CASE command, and any CASE command overrides any previously entered CAPS command.

Setting the Prefix Area Position

Use the Prefix command to specify if and where the prefix area is displayed:

| | |
|---------------------|--|
| PREFIX ON | The prefix area is displayed at the position last set |
| PREFIX OFF | The prefix area is not displayed |
| PREFIX LEFT | The prefix area is displayed to the left of the data |
| PREFIX RIGHT | The prefix area is displayed to the right of the data. |

See “Prefix Area Commands” on page 48 for a list of commands that can be entered in the prefix area.

Recovering Previously Deleted Data

Use the RECOVER command to restore lines of data that you previously deleted during this editing session:

```
RECOVER n
```

recovers the last *n* lines that were deleted (default 1).

Recovered lines are inserted after the top line.

Clearing Commands in Prefix Area

Use the RESET command to clear any outstanding commands that appear in the prefix area.

Splitting and Joining Lines (LE and VE Only)

Use the following commands to split data onto two or more lines, or combine data from two or more lines. Note that these commands are available only in multiple-record mode; you cannot use them when viewing a single record.

JOIN Joins two lines; either the line containing the cursor and the next line (which then overlays the line starting at the cursor position), or the top line with the following line at the current column position.

SPLIT Splits a line in two; either the line containing the cursor, at the cursor position, or the top line at the current column position.

SPLTJOIN Splits or joins lines, depending on the position of the cursor. If the cursor position in a line or the current column position is after the last non-blank character, a join operation is performed as described above; otherwise a split operation is performed.

Sending Library Member Contents to POWER (LE Only)

Use the SUBMIT command when editing a VSE library member to submit the member contents to the POWER reader queue.

Including the Contents of Another Library Member (LE Only)

Use the GETFILE command to insert the contents of another library member after the top line:

```
GETFILE member.type
```

adds the specified member (from the same library and sublibrary) to the member you are editing. If you do not specify a member name, a panel appears on which you can specify a library name, sublibrary name, and member name.

Editing another Library Member without Exiting (LE Only)

Use the EDIT command to suspend the current member edit and edit another library member:

```
EDIT member.type
```

opens the specified member (from the same library and sublibrary) for editing. If you do not specify a member name, a panel appears on which you can specify a library name, sublibrary name, and member name.

Additional Commands for Single-Record Editing

You can use the following commands only when editing in ZOOM (single-record) mode:

| | |
|---------------------|--|
| PUT | Saves data in a buffer for later retrieval. For example, <code>PUT col1-col2</code> saves the data from column <code>col1</code> to column <code>col2</code> of the current record in a buffer. You can later retrieve this data using the <code>GET</code> command. If you enter only one column number, the single character in that column is saved in a buffer. |
| GET | Retrieves the data previously stored in the buffer by the <code>PUT</code> command, and replaces the existing data, starting at the first byte of data displayed. For example, while browsing a tape, you can use <code>PUT</code> to copy a record into a buffer. Later, while editing a VSAM data set, you can use <code>GET</code> to get the data and add it to the data set. |
| MB CLIPBOARD | To see the data stored in the buffer, use the command <code>MB CLIPBOARD</code> . The length of the data is stored beginning with offset <code>X'C'</code> ; the data itself is stored beginning with offset <code>X'10'</code> . |
| COPY | Copies data in specified positions to the current position, replacing existing data starting at the first byte of data displayed. For example, <code>COPY col1-col2</code> copies the data from column <code>col1</code> to column <code>col2</code> of the current record to the current position, replacing the current data. |

Prefix Area Commands

To manipulate data in the file or data set, you can enter commands in the prefix area of the data display. The position of the prefix area is set with the `PREFIX` command (see "Setting the Prefix Area Position" on page 46).

Most of the prefix area commands can be performed on either a single line or a block of consecutive lines:

- To perform an operation on a single line, you enter the appropriate command code.
- To perform an operation on a block of consecutive lines, either enter the command code preceded or followed by the number of lines, or enter the appropriate block command code at the start and end lines of the block.

You can use the following prefix area commands when editing:

| | |
|-------------|--|
| A, F | Copy or move the target after (following) this line. |
| B, P | Copy or move the target before (preceding) this line. |
| C | Copy this line. |
| Cn | Copy <i>n</i> lines, starting at this line. |
| CC | Copy block. Use <code>CC</code> at the start and end of the block. |
| D | Delete this line. |
| Dn | Delete <i>n</i> lines, starting at this line. |
| DD | Delete block. Use <code>DD</code> at the start and end of the block. |

| | |
|------------------|--|
| G | Get from the buffer lines that were saved with the S prefix command. |
| I | Insert one blank line after this line. |
| In | Insert <i>n</i> blank lines after this line. |
| L | Translate the line to lower case. |
| Ln | Translate <i>n</i> lines to lower case. |
| LL | Translate block to lower case. Use LL at the start and end of the block. |
| M | Move this line. |
| Mn | Move <i>n</i> lines, starting at this line. |
| MM | Move block. Use MM at the start and end of the block. |
| R | Repeat this line once. |
| Rn | Repeat this line <i>n</i> times. |
| RR | Repeat block. Use RR at the start and end of the block. |
| RRn | Repeat block <i>n</i> times. Use RR <i>n</i> at the start and RR at the end of the block. |
| S | Save the line in a buffer. |
| Sn | Save <i>n</i> lines in a buffer. |
| SS | Save block in a buffer. Use SS at the start and end of the block. |
| U | Translate the line to upper case. |
| Un | Translate <i>n</i> lines to upper case. |
| UU | Translate block to upper case. Use UU at the start and end of the block. |
| > | Shift this line right one position. |
| >n | Shift this line right <i>n</i> positions. |
| >> | Shift block right one position. Use >> at the start and end of the block. |
| >>n | Shift block right <i>n</i> positions. Use >> <i>n</i> at the start and >> at the end of the block. |
| < | Shift this line left one position. |
| <n | Shift this line left <i>n</i> positions. |
| << | Shift block left one position. Use << at the start and end of the block. |
| <<n | Shift block left <i>n</i> positions. Use << <i>n</i> at the start and << at the end of block. |
| " | Duplicate this line once. |
| "n | Duplicate this line <i>n</i> times. |
| "" | Duplicate block. Use "" at the start and end of the block. |
| ""n | Duplicate block <i>n</i> times. Use "" <i>n</i> at the start and "" at the end of the block. |
| / | Position this line at the top of the screen (make it the current line). |

Saving and Exiting

Use the SAVE command to save changes you have made to the current data set or member, without leaving the edit window. You can also save the data set or member with a new name.

Use the EXIT command or function key to save changes and leave the edit window. Use the CANCEL command or function key to leave the edit window without saving changes.

VSAM Edit Restrictions and Considerations

The VSAM data set type affects the behavior of the VSAM Edit function as follows:

- In a KSDS, moving records has no effect. If you want to change the order of keyed records, change the key values.
- In an ESDS, you cannot move, delete, or change the length of existing records. If a record is compressed, you cannot alter the contents.

Edit

- In an RRDS, you cannot change the record length, or move existing records.
- If you delete a record in a RRDS or VRDS, the slot remains empty. Subsequent records do not change their slot number.

Altered and added records are highlighted in the display. Deleted records or free slots in RRDS or VRDS data sets are displayed as free slots and can be activated (undeleted) by entering the insert command in the prefix area (see “Prefix Area Commands” on page 48 for details of prefix area commands).

DITTO keeps only changed records in memory. If you run out of memory, the SAVE command may free some memory, which would let you continue the edit session.

Chapter 5. Processing Catalog Entries

You can use the following functions to process catalog entries:

Figure 33. DITTO Functions for Catalog Processing

| Purpose | Functions | |
|---|------------------|--------------------|
| Work with all catalog entries (VSAM and non-VSAM) | SCS | Catalog Services |
| Work with only VSAM catalog entries | VDL | VSAM Data Set List |

Printing a List of Catalog Entries

Under MVS or VSE, you can use the **SCS** function to print a list of catalog entries.

You can use the **VDL** function to print a list of only the VSAM entries in a catalog.

You can:

- Print a subset of the catalog entries based on a generic name
- Print a subset of the catalog entries based on entry type
- Sort the list by name, free space, creation date, expiration date, or (VSE or CMS) extent

The list includes the following information for each entry:

- Name
- Entry type

The following additional information is provided for VSAM data sets only:

- Primary volume
- Number of extents
- Share option values
- Creation and expiration dates
- Allocation information including:
 - The high allocated RBA
 - The high used RBA
 - The amount of free space
- VSE or CMS: information about a VSAM space listing:
 - Beginning cylinder and head
 - End cylinder and head
 - Relative track and number of tracks

Working with Catalog Entries in Full-Screen Mode

In full-screen mode, you can use the **SCS** or **VDL** function to work with catalog entries. These functions are very similar, but have the following differences:

- **SCS** is available only under VSE and MVS.
- **VDL** works with only VSAM catalog entries.
- Under VSE and CMS, **VDL** only lets you view or print data; with **SCS** (or **VDL** under MVS), you can also manage catalog entries.

To work with catalog entries, select the SCS option from the DITTO Primary Function menu, select the VDL option from the VSAM Function menu, or enter VDL or SCS on the command line of any panel in DITTO.

A selection panel appears. Figure 34 on page 53 and Figure 35 on page 53 show the SCS selection panels for MVS and VSE.

| Process | View | Options | Help |
|--|---------|------------------------|---------|
| DITTO/ESA for MVS | | SCS - Catalog Services | |
| Data Set _____ | | | |
| Data set name ** _____ | | Catalog ID _____ | |
| Object type ALL _____ | ALL | CLUSTER | NONVSAM |
| | ALLVSAM | DATA | OAM |
| | ALIAS | GDG | PGSPC |
| | AIX | INDEX | PATH |
| Catalog Service _____ | | | |
| Command LIST _____ | ALTER | DELETE | PRINT |
| | DEFINE | LIST | |
| Sort type NAME _____ | NAME | EDATE | FREESP |
| | | | DATE |
| Command ==> | | | |
| F1=Help F3=Exit F10=Actions F11=CRetrieve F12=Cancel | | | |

Figure 34. Catalog Services Selection Panel (MVS)

| Process | View | Options | Help |
|--|---------|------------------------|--------|
| DITTO/ESA for VSE | | SCS - Catalog Services | |
| Data Set _____ | | | |
| Catalog ID VSAM.MASTER.CATALOG _____ | | Space VOLID _____ | |
| Data set name ** _____ | | | |
| Object type ALL _____ | ALL | DATA | PATH |
| | ALLVSAM | INDEX | SPACE |
| | AIX | NONVSAM | UCAT |
| | CLUSTER | | |
| Catalog Service _____ | | | |
| Command LIST _____ | ALTER | DELETE | PRINT |
| | DEFINE | LIST | |
| Sort type NAME _____ | NAME | EDATE | FREESP |
| | DATE | EXTENT (space listing) | |
| Command ==> | | | |
| F1=Help F3=Exit F10=Actions F11=CRetrieve F12=Cancel | | | |

Figure 35. Catalog Services Selection Panel (VSE)

Note: The VDL panel offers only those object types that apply to VSAM data sets. Under VSE or CMS, it also offers only the List and Print commands.

You can use the commands that are listed on the panel as follows:

Alter To alter the definition of a single catalog entry, type the catalog name and data set name, and enter **Alter** in the Command field.

A panel appears, on which you can change parameters for the entry, as described in “Parameters for a Catalog Entry” on page 58.

- Define** To define a new catalog entry using an existing catalog entry as a model, type the catalog name and data set name for the *existing* entry, and enter `Define` in the Command field.
- A panel appears, on which you enter the data set name of the new entry. You can also change the parameters (copied from the existing entry), as described in “Parameters for a Catalog Entry” on page 58.
- To define a new catalog entry *without* an existing catalog entry as a model, type the catalog name, data set name, and object type for the new entry, and enter `Define` in the Command field.
- You are prompted for various parameters, as described in “Parameters for a Catalog Entry” on page 58.
- Delete** To delete a single catalog entry, type the catalog name and data set name, and enter `Delete` in the Command field.
- Print** To print a list of catalog entries, type the catalog name, under VSE or CMS optionally type the name of the volume that contains a space, and enter `Print` in the Command field. You can sort the list by data set name, free space, creation date, expiration date, or (VSE or CMS) extent. You can also limit the list by typing a generic name or object type.
- List** To list the parameters of a single catalog entry within SCS, type the catalog name, and data set name, and enter `List` in the Command field.
- To display a list of catalog entries, type the catalog name, under VSE or CMS optionally type the name of the volume that contains a space, and enter `List` in the Command field. You can sort the list by data set name, free space, creation date, expiration date, or (VSE or CMS) extent. You can also limit the list by typing a generic name or object type.
- You can perform several functions directly from the displayed list, as described in the next section.

Working with a List of Catalog Entries

When you enter the `List` command on the SCS or VDL selection panel, a list of catalog entries appears. Figure 36 on page 55 shows the SCS primary panel.

```

Process  View  Options  Help
-----
DITTO/ESA for MVS          SCS - Catalog Services          Line 1 of 67

Catalog ID 'CATALOG.LPS'
--- Data Set Name --- sorted by NAME    --- Entry Prim M Created Expires
1...5...10...15...20...25...30...35...40... type volume V YYYY.DDD YYYY.DDD
CATALOG.LPS                UCAT  CCA01  2000.291
DVG.NFTP220.ADVGINSO      NVSAM P390DX 1997.091
DVG.NFTP220.ADVGMACO      NVSAM P390DX 1997.091
DVG.NFTP220.ADVGMODO      NVSAM P390DX 1997.091
DVG.NFTP220.AVGMOD2      NVSAM P390DX 1997.091
DVG.NFTP220.AVGMMSG0      NVSAM P390DX 1997.091
DVG.NFTP220.AVGPNL0      NVSAM P390DX 1997.091
DVG.NFTP220.AVGSACO      NVSAM P390DX 1997.091
DVG.NFTP220.AVGSAMO      NVSAM P390DX 1997.091
DVG.NFTP220.AVGSARO      NVSAM P390DX 1997.091
DVG.NFTP220.CHECKPNT.FILE KSDS          1997.092
  DVG.NFTP220.CHECKPNT.FILE.DATA DATA SCPMV5 1997.092
  DVG.NFTP220.CHECKPNT.FILE.INDEX INDEX SCPMV5 1997.092

Command ==> _____ Scroll PAGE
F1=Help F2=List F3=Exit F4=Process F5=RgtLeft F6=RFind F7=Bkwd F8=Fwd
F9=Swap F10=Actions F11=CRetrieve F12=Cancel

```

Figure 36. Catalog Services (Primary Panel)

If an IDCAMS error occurs (for example, if a data set is not found), ERROR appears in the ENTRY TYPE field.

By using the RgtLeft function key, you can switch between two display formats to see all the information you would get on a full-width printout. Figure 37 shows the second display format (with the Process pull-down menu open).

```

Process  View  Options  Help
-----
1. Alter
2. Define
3. Delete
4. List      F2
5. Browse
6. View
7. Edit
8. Extents
9. Print
10. Update

11. AMS message
12. Refresh
13. Exit     F3
14. Exit DITTO  LE.DATA      4,3  1    160    160    80
      LE.INDEX  4,3  1     40     4     36

SCS - Catalog Services          Line 1 of 67

ted by NAME    --- Shr Ext -----Kilobytes-----
...30...35...40... opt nbr Hi-Alloc Hi-Used Free-Spc
                3,3  1    5600    560    5600

Command ==> _____ Scroll PAGE
F1=Help F2=List F3=Exit F4=Process F5=RgtLeft F6=RFind F7=Bkwd F8=Fwd
F9=Swap F10=Actions F11=CRetrieve F12=Cancel

```

Figure 37. Catalog Services (Alternative Display, MVS)

Note: In SCS under VSE, the Directory and Status options are also available. The Alter, Define, Delete, and Amsmsg options are not available in VDL under VSE or CMS.

To move forward or backward through the list, use the Fwd or Bkwd function key, or use the locate command. You can also enter a value in the Scro11 field. (In DITTO, all full-screen list functions and full-screen browse functions use the same basic methods for locating data. These methods are described in Chapter 3, “Browsing Data in Full-Screen Mode” on page 31.)

If you press the Process function key, you see a pull-down menu that lists the following options:

| | |
|------------------|--|
| Alter | Lets you alter the parameters for the selected entry. A panel appears, on which you can change parameters for the entry, as described in “Parameters for a Catalog Entry” on page 58. |
| Define | Lets you define a new entry that is similar to the selected entry. A panel appears, on which you enter the name of the new entry. You can also change the parameters (copied from the existing entry), as described in “Parameters for a Catalog Entry” on page 58. |
| Delete | Deletes the selected entry. |
| List | If the selected entry is a catalog or space, invokes SCS or VDL for that catalog or space; otherwise, lists the parameters for the selected entry. |
| Browse | If the selected entry is an OAM object collection, catalog, or space, invokes the appropriate DITTO list function (ODL , SCS , or VDL); otherwise, invokes the appropriate DITTO browse function (VB , DVB , or DB). |
| Edit | Invokes the appropriate edit function (VE or DTE). |
| View | Invokes APB for NONVSAM and VSE SAM ESDS data sets. |
| Directory | If the selected entry is a VSE library, invokes the LDL function. A library has type NONCI in the SCS or VDL list (but not all entries with type NONCI are libraries). |
| Status | If the selected entry is a VSE library, invokes the LSL function. |
| Print | Invokes the appropriate print function (VP or DP). |
| Update | Invokes the appropriate DITTO update function (VU or DU). |
| Amsmsg | Displays any IDCAMS messages associated with the current task. |
| Refresh | Replaces the current list with a new list taken from the catalog. The new list incorporates any changes that you have made, and also any changes made by other users since you first displayed the list or last refreshed it. Note: For performance reasons, the current list is kept until you refresh it or terminate DITTO. For example, if you quit SCS and later (in the same DITTO session) call SCS with the same parameters, you see the list from the previous SCS session. |
| Exit | Ends the VDL or SCS function. |

In general, there are two ways to perform any of these functions:

- You can use the Process pull-down menu:
 1. Move the cursor to the entry that you want to work with. (If the cursor is outside the list area, you will work with the first entry in the list.)
 2. Press the Process function key. The selected line is highlighted in another color.
 3. In the entry field of the pull-down menu, type the number associated with the function and press Enter.
- You can type a command:
 1. Move the cursor to the *beginning* of the name of the entry that you want to work with.
 2. Type the command (for example, browse) over the name. You can type an equal sign (=) to perform the same command for more than one entry.

The Process pull-down menu options work differently for different types of catalog entries. Figure 38 shows the functions that are available.

| <i>Figure 38. Functions Available for Different Types of Entries</i> | | | | | | | | |
|--|-------|--------|--------|------------|---------------|--------------------|-------|---------------|
| | Alter | Define | Delete | List | Browse / View | Directory / Status | Print | Update / Edit |
| AIX | yes | yes | yes | yes | VB | | VP | VU / VE |
| MVS: Alias | | yes | yes | yes | | | | |
| Cluster | yes | yes | yes | yes | VB or DVB | | VP | VU / VE |
| MVS: GDG | | yes | yes | yes | | | | |
| VSE: Library | yes | yes | yes | yes | | LDL / LSL | | |
| VSE: Non-VSAM | yes | yes | yes | yes | | | | |
| CMS: Non-VSAM | | | | yes | | | | |
| MVS: Non-VSAM | yes | yes | yes | yes | DB / APB | | DP | DU / DTE |
| MVS: OAM | | | | yes | ODL | | | |
| Path | yes | yes | yes | yes | VB | | VP | VU |
| MVS: PGSPC | | | | yes | | | | |
| VSE: SAMDS | yes | yes | yes | yes | VB / APB | | VP | VU / VE |
| VSE/CMS: Space | | | | SCS or VDL | | | | |
| UCAT | | | | SCS or VDL | | | | |
| Notes: | | | | | | | | |
| <ul style="list-style-type: none"> • A blank space means that the function is not available. • The word “yes” means that the function is available. • The name of a DITTO function means that this DITTO function is invoked. • “SCS or VDL” means that the SCS or VDL function is invoked again to list the entries in a catalog or space. • You can delete a cluster, but the deletion attempt may fail if the cluster is large or complicated (for example, with more than 1 AIX or more than 20 volsers per component). | | | | | | | | |

Parameters for a Catalog Entry

You can display, define, or alter IDCAMS parameters for a catalog entry as follows:

- Select List to display parameters without changing them.
- Select Alter to display parameters and change them.
- Select Define to create a new entry.

When you select one of these functions, IDCAMS parameters for the selected entry are displayed. Different IDCAMS parameters are displayed for different types of entries.

Figure 39 shows the panel that appears. You can scroll up and down the list of parameters. Only the most commonly used IDCAMS parameters are shown. To change any other IDCAMS parameters, you must use IDCAMS itself. For more information about IDCAMS parameters, see *VSE/VSAM Commands* or *DFSMS/MVS® Access Method Services for Integrated Catalog Facility*.

```

Process   View   Options   Help
-----
DITTO/ESA for MVS          SCS - Catalog Services

Data set name. . . 'AIXHLS.ITT.ESDS.AIX2'
Catalog ID . . . . 'SYS1.DUTCAT.VDUTVOL'

More:  -+

AIX Data Allocation:

Allocation unit. . . . TRK REC, BLK, TRK, or CYL

Space:      Primary . . 1          Secondary . . 0
Record size: average . . 4086      maximum . . 32600

Free space: % of CI . . 0          % of CA . . 0

Command ==>
F1=Help  F3=Exit  F5=Statistics F7=Bkwd  F8=Fwd  F10=Actions  F11=CRetrieve
F12=Cancel
    
```

Figure 39. Catalog Parameters Panel

Parameters for a VSAM Entry

If you select a VSAM entry (AIX, CLUSTER, or PGSPC), you can scroll through several panels showing:

- The creation date (yyyy.ddd)
- The expiration date (yyyy.ddd)
- Under MVS, the SMS attributes
- VSAM associations (cluster, AIX, and path)
- The VSAM data type (KSDS, ESDS, RRDS, VRDS, SAMDS, or NONCI)
- The control interval size
- The buffer space to be allocated at open time
- The share options
- Reuse (Y or N)
- Recovery (Y or N)
- Spanned (Y or N)

- VSE: Compressed (Y or N)
- Erase (Y or N)
- The space unit for allocation (REC, BLK, KB, MB, TRK, or CYL)
- The number of units for primary and secondary allocation
- The average and maximum record size
- Free space, as a percentage of the control interval and control area
- Up to five volume serials and device types
- The key length
- The position of the key within the record (starting with 1)
- Whether the key must be unique (Y or N)
- Up to five key ranges (specified as hexadecimal numbers)
- Imbed (Y or N)
- Replicate (Y or N)

Parameters for a Path

If you select a path, you can see:

- The creation date (*yyyy.ddd*)
- The expiration date (*yyyy.ddd*)
- The alternate index
- The cluster

VSE: Parameters for a Non-VSAM Entry

If you select a non-VSAM entry under VSE, you can see up to five volume serials and device types.

MVS: Parameters for a Non-VSAM Entry

If you select a non-VSAM entry under MVS, you can see:

- The creation date (*yyyy.ddd*)
- The expiration date listed in the catalog (*yyyy.ddd*)
- Up to five volume serials and device types
- The SMS attributes

Note: If you change the expiration date for a non-VSAM entry with **SCS**, this change does not affect the expiration date in the disk VTOC or tape header. Only the catalog entry is changed.

If the data set is a GDG member or has aliases, a second panel appears. On this panel, you can see:

- The GDG base
- The alias or aliases

MVS: Parameters for a GDG Base

If you select a GDG base, you can see:

- The creation date (*yyyy.ddd*)
- The expiration date (*yyyy.ddd*)
- The maximum number of GDG members
- The names of the first five GDG members

MVS: Parameters for an OAM Object Collection

If you select an OAM object collection, you can see:

- The creation date (*yyyy.ddd*)
- The expiration date (*yyyy.ddd*)
- The SMS attributes
- The directory token

MVS: Parameters for an Alias

If you select an alias, you can see the user catalog that the alias points to.

Statistics for VSAM Data Sets

If you press the Statistics function key on a catalog parameters display, VSAM statistics are displayed for every extent on every volser for the selected data set. Figure 40 shows an example of the statistics panel.

```

Process  View  Options  Help
-----
DITTO/ESA for MVS          SCS - Catalog Services

Data set name. . . 'MHEIHS.ITT.ESDS.AIX2'
Catalog ID . . . . 'SYS1.DUTCAT.VDUTVOL'

More: +-

Data Extents:

EXT      BEGIN ----- END      -- high --
# VOLSER  CYL-HD  CYL-HD RELTRK,NUMTRKS --low RBA- -high RBA- -used RBA-
DUTVOL
1  --"--  222  0  222  0  3330,1          0  43008  43008
. . . . .

Index Extents:

Command ==>
F1=Help F2=Browse F3=Exit F4=Process F5=List F7=Bkwd F8=Fwd F10=Actions
F11=CRetrieve F12=Cancel
    
```

Figure 40. Catalog Statistics Panel

Chapter 6. Processing Tapes

You can use the following functions to process or query tapes:

Figure 41. DITTO Functions for Tape Processing

| Purpose | Functions |
|---|--|
| Get information about tape labels or contents | TLB Tape Label Display TMP Tape Map |
| Browse tape blocks at your terminal | TB Tape Browse |
| Look at a tape block on the system console | TCN Tape to Console |
| Update physical tape records | TRL Tape Record Load TU Tape Update |
| Print tape records in character or hexadecimal format | TP Tape Print |
| Print VSE SYSLST tapes | TFA Print SYSLST Tape Type A TFD Print SYSLST Tape Type D |
| Copy data from tape to tape | TT Tape to Tape Copy TTR Tape to Tape Reblocked |
| Compare tapes to see if their contents are identical | TTC Tape to Tape Compare |
| Search for a specific string on a tape | TRS Tape Record Scan |
| Position the tape at a specific file or physical record | BSF Backward Space File BSR Backward Space Record FSF Forward Space File FSR Forward Space Record |
| Rewind the tape | REW Tape Rewind RUN Tape Rewind-Unload |
| Write one or more tape marks | WTM Write Tape Mark |
| Erase tape contents (data security erase) | ERT Erase Tape |
| Initialize a tape in EBCDIC or ASCII format | INT Initialize Tape |
| Copy labeled tape files | TFT Tape File to Tape File TLT Tape to Labeled Tape |
| Copy tape records to a file | TF Tape to CMS File TS, TQ Tape to Sequential Data TV Tape to VSAM |
| Copy tape records to a library member | TL Tape to Library Member |
| Copy tape records to cards | TC Tape to Card |
| Copy tape records to or from a REXX stem variable | TX Tape to REXX Variable XT REXX Variable to Tape |
| Create data in a tape file | BT Create Tape File |

Specifying Tape Input or Output

With DITTO tape functions, you can process tape data in either of the following ways:

Basic Tape With basic tape processing, you work with tape files and tape blocks. The first tape mark after a data record or after the leading tape mark is treated as the end of a file. Basic tape functions cannot process multivolume input tape files. No tape label processing is performed; tape label sets are processed as ordinary tape files.

You can move the tape forward or backward by a number of physical records or a number of tape files. Tape input is read from the current position of the input tape, and tape output is written to the current position of the output tape.

Notes:

1. For output tapes, you can position the tape before the last tape mark of a file if you want to extend the tape file.
2. To process a labeled input tape without label processing, position the tape at the first data record (for example, by using the **FSF** command to skip the tape label file).
3. If the first record read is a tape mark, DITTO considers it a leading tape mark and continues. To copy a null file correctly, position the tape in front of the tape mark, which will then be treated as a leading tape mark. To copy consecutive files (including null files) from a multifile input tape, issue a backspace record (**BSR**) command before each copy command.
4. At the end of tape, DITTO writes two tape marks to indicate the end of volume, unloads the full tape, and requests the mounting of another volume on the tape output device. You can thus create an unlabeled multivolume tape file.
5. MVS: for tape functions that cross tape marks and for tape output functions, BLP or NL processing is recommended.

Labeled Tape For labeled tape processing, you specify a file ID. An input tape must be positioned in front of the specified label set. (If the file IDs do not match, the function ends with an error.) An output tape must be positioned at the beginning of a tape or immediately after a labeled file.

DITTO processes SL and AL (ANSI) labels. On input, the label format is recognized automatically. For output, use the SET option TAPELBL to select the label type.

Labeled tape processing depends on the operating system, as follows:

MVS Labeled tape processing is specified when the tape is opened (in the JCL or during interactive allocation). Only the following functions are enabled for output label processing: BT, CT, QT, ST, and VT.

VSE For functions other than **TLT**, labels are processed by logical input/output control system routines (DTFMT). For a multivolume tape in batch mode, specify alternate assignments to avoid messages prompting for the new volumes.

CMS DITTO processes all labels. Multivolume input and output files are supported. Any record format options specified are used for HDR2 label setup.

Note: Not all combinations of DITTO keyword values are supported by the underlying operating environment. For information on supported combinations, see your operating environment documentation.

MVS: Specifying Tape Input or Output with JCL

Under MVS, you can specify tape input or output with JCL. In the JCL, you supply DD statements for the tape data sets that you want to access. You use the ddnames from these DD statements in DITTO commands that have the parameter `INPUT=ddname` or `OUTPUT=ddname` (or both).

The first DITTO function that uses a tape opens the tape. If you use functions that are not enabled for output label processing, the tape remains open until you rewind and unload it (**RUN** function) or end the job (**EOJ** function). If you use a function that is enabled for output label processing (BT, CT, QT, ST, or VT), the tape is rewound and unloaded after the file is closed and the function has completed.

There are four ways to access a tape:

- To access a tape in read only mode, specify a DITTO command with the keyword `INPUT=ddname` as the *first* DITTO command that uses the tape.

The tape should be write-protected. The tape is opened using `OPEN=INPUT`.

- To access a tape in read/write mode, specify a DITTO command with the keyword `OUTPUT=ddname` as the *first* DITTO command that uses the tape.

The tape should not be write-protected, and the operator might need to confirm that the tape can be used. The tape is opened using `OPEN=INOUT`, so that the tape can be used for output if desired but labels are not rewritten during open.

- To access a scratch tape (a non-specific volume request), specify a DD statement without the `VOL` keyword (or without the `SER` or `REF` subparameter of the `VOL` keyword).

The tape should not be write-protected, and the operator might need to confirm that the tape can be used. The tape is opened using `OPEN=OUTIN`, so that the tape can later be used for input if desired.

- To force bypass label processing (BLP), specify a tape command with the `LABEL=BLP` keyword as the *first* DITTO command that uses the tape. You must also specify deferred mount (`UNIT=(devicenumber, ,DEFER)`) in the DD statement. For information about when BLP is available, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

- To write an SL tape using a function enabled for output label processing, specify `LABEL=(n,SL)` in the JCL with a `DISP=NEW`. The tape file will be closed and the tape rewound and unloaded when the function completes. Tape SL output and tape non-SL functions cannot be used in the same execution of DITTO. To append to the SL file, specify `DISP=MOD` in the JCL.

Tape

For an output tape, you can optionally specify `DENS=mm`, where *mm* is a tape mode code shown in “Tape Density and Mode Values” on page 67.

```
//S1      EXEC PGM=DITTO
//TAPE1   DD DISP=(OLD,KEEP),DSN=DUMMY,LABEL=(,BLP),
//        UNIT=480,VOL=SER=DITTP1
//TAPE2   DD DISP=(NEW,KEEP),DSN=DUMMY,LABEL=(,BLP),
//        UNIT=481,VOL=SER=DITTP2
//TAPE3   DD DISP=(NEW,KEEP),DSN=DUMMY,LABEL=(,BLP),
//        UNIT=482
$$DITTO TP INPUT=TAPE1
$$DITTO REW OUTPUT=TAPE2
$$DITTO TT INPUT=TAPE2,OUTPUT=TAPE3
$$DITTO EOJ
/*
//
```

Figure 42. Example of Allocating Tapes in Batch Mode under MVS

Figure 42 shows three ways to allocate tapes:

- TAPE1 is accessed in read only mode.
- TAPE2 is accessed in read/write mode.
- TAPE3 is accessed as a scratch tape.

VSE: Specifying Tape Input or Output with JCL

Under VSE, you can specify tape input or output with JCL as follows:

1. Specify `INPUT=` or `OUTPUT=` followed by the device address or programmer logical unit of the tape unit.
2. For an output tape, optionally specify `DENS=mm`, where *mm* is a tape mode code shown in “Tape Density and Mode Values” on page 67.
3. If you want to perform labeled tape processing (and the function you are using supports labeled tape processing), you also identify the file you want to use as follows:
 - In the JCL, specify a TLBL statement for the file.
 - On the DITTO tape-processing command, specify `FILEIN=` or `FILEOUT=` followed by the name of the TLBL statement. If you used the default TLBL name (TAPEIN for tape input or TAPEOUT for tape output), you do not need a FILEIN or FILEOUT statement.
4. To work with an automated tape library, use the VSE LIBSERV command to specify the volumes to be used (required in batch mode).

MVS: Specifying Tape Input or Output without JCL

Under MVS, you can specify tape input or output without JCL as follows:

1. Specify a ddname for the tape unit.
2. For an output tape, optionally specify a tape mode code, as shown in “Tape Density and Mode Values” on page 67. In line mode (or command mode with the parameter syntax), the tape mode code follows the ddname, separated by a hyphen (for example, TAPE1-08). By default, DITTO does not change density, and uses buffered write mode where applicable.
3. If the ddname is not allocated, you are asked for allocation information.

The first DITTO function that uses a tape opens the tape. If you use functions that are not enabled for output label processing, the tape remains open until you rewind and unload it (**RUN** function) or end the job (**EOJ** function). If you use a function that is enabled for output label processing (BT, CT, QT, ST, or VT), the tape is rewound and unloaded after the file is closed and the function has completed.

When you allocate a tape, you are asked for the following information:

| | |
|------------------------|---|
| Volume serial | <p>The actual volser (which is verified if label processing is requested), or an external number to be used for mounting the tape if you do not know the actual volume serial (which requires BLP). For example, you might use TLB to find out the actual volume serial number.</p> <p>To access a volume set, specify up to five volsers in parentheses, for example: (DIT005 DIT006 DIT007). Volume sets are supported by TLB, TLT, and basic tape output functions.</p> <p>To access a scratch tape (a non-specific volume request), do not enter a volser. The tape should not be write-protected, and the operator might need to confirm that the tape can be used. The tape is opened using OPEN=OUTIN, so that the tape can later be used for input if desired.</p> |
| Unit | Tape unit number. You can respond with a device number, device type (generic name), or group name (symbolic name). |
| Label value | <p>SL for standard label processing (the default), AL for ANSI label processing, BLP for bypass label processing, or NL for no labels.</p> <p>If DITTO is running APF-authorized, BLP is set internally for some functions if the user has the appropriate authority. BLP is reset when the function ends.</p> |
| Data set name | The name of the tape data set that you want to work with, if label processing is used. |
| Sequence number | The relative position of a data set on the tape. |
| Expiration date | The expiration date for the data set. |
| Open for write | Specify YES to access a tape in read/write mode. The tape should not be write-protected, and the operator might need to confirm that the tape can be used. The tape is opened using |

OPEN=INOUT, so that the tape can be used for output if desired but labels are not rewritten during open.

If the DITTO session abends, you might need to use the TSO FREE command to remove any DITTO allocations.

Note: You can also allocate tapes outside DITTO. There are two reasons why you might want to do this:

- In command mode, keywords for tape allocation are not available. You can allocate a tape if you use the parameter syntax (in which you specify parameters in the required order with no keywords). Unless you are thoroughly familiar with the order of allocation prompts, you will probably find it more convenient to allocate tapes outside DITTO.
- In full-screen mode or line mode, you might want to use allocation parameters that you cannot specify within DITTO.

VSE: Specifying Tape Input or Output without JCL

Under VSE, you can specify tape input or output without JCL as follows:

1. Specify the device address of the tape unit.
2. To work with any free automated tape library device in full-screen mode or line mode, specify ATLC for the tape unit, where *c* is an alphanumeric character. You are prompted for the external volume serial number as it is known by the Tape Library Dataserver. You can specify up to 5 volsers. You are also prompted for the name of the library that contains the volumes, and whether the volumes should be mounted for read or read/write.

DITTO reserves the device for the partition until you release it (**RUN** function with the *re/lease* parameter) or end the job (**EOJ** function).

3. Optionally specify a tape mode code. In line mode (or command mode with the parameter syntax), the tape mode code follows the device address, separated by a hyphen (for example, 181-08). The tape mode code is required when you work with a 7-track tape. The tape mode code does not apply to tape positioning commands, only to commands that actually read tape input or write tape output. For more information about tape mode codes, see "Tape Density and Mode Values" on page 67.
4. If you want to perform labeled tape processing (and the function you are using supports labeled tape processing), specify INFILE= or OUTFILE= with the file ID of the file you want to use.

CMS: Specifying Tape Input or Output

Under CMS, you can specify tape input or output as follows:

1. Specify the device address of the tape unit.
2. Optionally specify a tape mode code. In line mode (or command mode with the parameter syntax), the tape mode code follows the device address, separated by a hyphen (for example, 181-08). The tape mode code is required when you work with a 7-track tape. The tape mode code does not apply to tape positioning commands, only to commands that actually read tape input or write tape output. For more information about tape mode codes, see "Tape Density and Mode Values" on page 67.

3. If you want to perform labeled tape processing (and the function you are using supports labeled tape processing), specify the file ID of the file you want to use.

Tape Density and Mode Values

Figure 43 and Figure 44 show the density and mode values that are available.

| <i>Figure 43. Density/Mode Values for Tapes That Are Not 7-Track</i> | |
|--|--|
| Tape Mode (mm) | 3480 / 3490 Cartridge Tapes |
| 08 28 00 or BU 20 or IM | Compaction (IDRC), buffered write mode Compaction (IDRC), immediate write mode No compaction, buffered write mode No compaction, immediate write mode |
| Tape Mode (mm) | 3400 Tapes |
| C8 * C0 * D0 * | 800 bpi, dual density 1600 bpi, dual density 6250 bpi, dual density |
| Tape Mode (mm) | 3424 / 9348 Tapes |
| 42 * 62 * C2 * , 00 or BU E2 * , 20 or IM | 1600 bpi, buffered write mode 1600 bpi, immediate write mode 6250 bpi, buffered write mode 6250 bpi, immediate write mode |
| Tape Mode (mm) | 9346 / 9347 Streamer Tapes |
| 50 or LL 60 or LS 90 or HL 30 or HS | IBM 9347, low speed, long gap IBM 9347, low speed, short gap IBM 9347, high speed, long gap IBM 9347, high speed, short gap |
| 00 to FE | IBM 9346, forced streaming mode |

| <i>Figure 44. Tape Conversion Mode Values for 7-Track Tapes</i> | | | | |
|---|----------------------|-----------------------------------|-------------------------------|--------------------------------|
| Tape Mode (mm) | Density (bpi) | Parity | Translation | Conversion |
| 10 * 20 * 28 * 30 * 38 * | 200 | Odd Even Even Odd Odd | Off Off On Off On | On Off Off Off Off |
| 50 * 60 * 68 * 70 * 78 * | 556 | Odd Even Even Odd Odd | Off Off On Off On | On Off Off Off Off |
| 90 * A0 * A8 * B0 * B8 * | 800 | Odd Even Even Odd Odd | Off Off On Off On | On Off Off Off Off |

Notes:

1. For mm values marked with an asterisk (*), the tape must be positioned at the load point.
2. If the mm value is omitted, the value specified in JCL or the system default is used.
3. VSE: for labeled tape processing functions other than **TLT**, you cannot set the density within DITTO. (DITTO gets control after OPEN processing, so the tape is no longer positioned at the load point and the density cannot be changed.)
4. VSE: for IBM 9347 tape units, an attention routine SETMOD command is recommended for better performance.

VSE: Permanent or Temporary Tape Assignment

Under VSE, tapes can be assigned temporarily (for the duration of one function only) or permanently (for the duration of the DITTO session), as follows:

- In full-screen mode or line mode, use SET TAPEASGN=TEMP or SET TAPEASGN=PERM to temporarily or permanently assign tapes. The default supplied by IBM is TAPEASGN=TEMP.
- In batch mode, tapes are always assigned for the duration of the job.
- In command mode, you leave DITTO immediately after a command is executed (so the duration of a function and the duration of a session are the same).

Tape Block Size Considerations

DITTO can process input and output tape blocks that are larger than 64KB. The theoretical limit for a tape block is 10MB.

If you use tape blocks that are larger than 64KB (32KB for labeled tape processing), be aware that other software may not be able to process such large tape blocks. The hardware that you are using may also limit the block size.

Positioning a Tape

To position a tape, use these DITTO functions:

- **FSR** or **BSR** moves the tape forward or backward a number of physical records.
- **FSF** or **BSF** moves the tape forward or backward a number of files.
- **REW** rewinds the tape.
- **RUN** rewinds the tape and unloads it. Under MVS, this function closes the tape.

MVS: You can also specify the data set sequence number in the tape allocation.

How Tapes are Organized

To correctly position a tape, you should know the format of the tape. Figure 45 on page 70 shows the format of IBM standard labeled tapes.

A *tape file* consists of the data between two *tape marks* (or between the load point and the first tape mark). Each data set in the figure consists of three tape files:

the header labels, the data, and the EOF (end of file) labels. To move forward by one data set, you would move forward by three tape files.

The first file on the tape consists of the volume label (VOL1) and the header labels (HDR*n*) defining the first data set.

Two tape marks after the end of file (EOF) label indicate the end of the last data set; on NL tapes, two tape marks indicate the end of the data on the tape. This *double tape mark* is also known as an EOVS (end of volume) indication. On multivolume labeled tapes, a tape mark after the EOVS label indicates the end of the data on this volume. To write tape marks yourself, use **WTM** (Write Tape Mark).

When you open a tape using standard label processing under MVS, the tape is positioned after the first header label set. Your first tape input or tape output function reads or writes the first record of the first data set. When you open a tape using bypass label processing (BLP) under MVS, the tape is positioned at the load point or begin-of-tape (BOT).

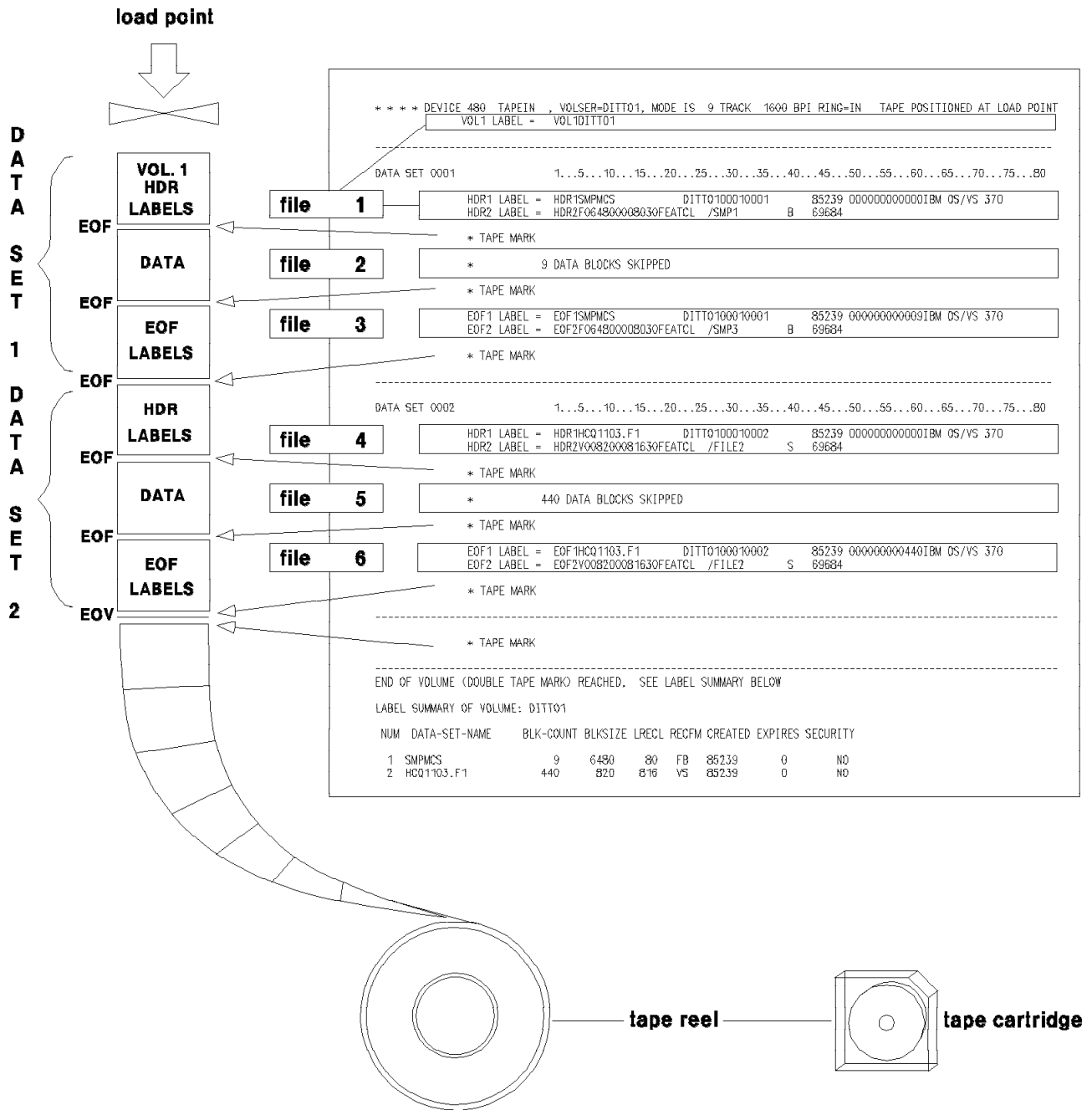


Figure 45. Layout of a Tape with IBM Standard Labels

Specifying the Number of Tape Files to Process

Several DITTO functions let you specify the number of tape files that you want to work with.

- You can accept the default (to process one tape file).
- You can specify a number of tape files from 1 to 99 999.
- With some functions, you can specify EOV to process all files until the end-of-volume (double tape mark or EOV label).

- With the TT function, you can specify EOT to copy until the physical end-of-tape. Continuation volumes are not requested.
- With some functions, you can specify EOD to process all files until a specified delimiter is encountered. First, use the **SET** function to set an EOD delimiter, then invoke a tape function with EOD as the number of files.

Some tapes (such as program installation tapes) may have multiple tape marks between files. To make sure that all files are processed or to process input tapes beyond the end-of-volume, specify EOD or a number greater than the number of files on the tape. To prevent a tape from running off the reel, DITTO stops if it encounters a data check after a tape mark. If reading records does not produce a data check, DITTO processes all records until the physical end of the tape is reached.

Processing Tapes with Data-Check Errors

In full-screen mode or line mode, you can process input tapes that have data-check errors. When a data-check error is found, DITTO lets you change, accept, or bypass the tape block containing the error.

- In full-screen mode, you can browse the block. You are prompted for what to do next with the block.
- In line mode, you are prompted for what to do next with the block.

The blocks that you change are printed, so you have a log of any changes you make.

Finding Out What a Tape Contains

If a tape is labeled (or if you want to find out if it is labeled) use the **TLB** function to print the tape labels. This allows you to determine the volume serial number of the tape, or to determine which data sets are on the tape.

MVS: you can use **TLB** even if you do not have access to the data on the tape, because only the tape labels are read. To mount and process tapes with unknown volsers and labels, **TLB** turns on BLP internally. If your installation does not allow BLP usage but DITTO is running APF-authorized, DITTO forces BLP for this function; for security reasons, the tape is freed (unallocated) when TLB ends.

Figure 46 on page 72 shows an example of the default output from **TLB**. You can also request a shorter report that lists only the VOL1 label and the header labels for the first data set.

```

DITTO/ESA for MVS
* * * * Device 384, T2, VOLSER=DUTSUL, 18 Track 38K-BPI, Uprotect
VOL1 label = VOL1DUTSUL
09/29/1998 (1998-273) 18:40 Page 2
* * * *

-----
Data set 0001      1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
HDR1 label = HDR1DUTPLIB      DUTSUL00010001      9017600000000000000000IBM OS/V5 370
HDR2 label = HDR2V031400313630PESENC /S1      S      64586
UHL1 label = UHL1This is User Header Label 1
* Tape mark
*
* 622 data block(s) skipped
* Tape mark
EOF1 label = EOF1DUTPLIB      DUTSUL00010001      90176000000000000000622IBM OS/V5 370
EOF2 label = EOF2V031400313630PESENC /S1      S      64586
UTL1 label = UTL1 This is User Trailer Label 1
* Tape mark

-----
Data set 0002      1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
HDR1 label = HDR1DUTLOAD      DUTSUL00010002      9017600000000000000000IBM OS/V5 370
HDR2 label = HDR2V061640616030PESENC /S2      S      64586
UHL1 label = UHL1 This is User Header Label 2
* Tape mark
*
* 122 data block(s) skipped
* Tape mark
EOF1 label = EOF1DUTLOAD      DUTSUL00010002      90176000000000000000122IBM OS/V5 370
EOF2 label = EOF2V061640616030PESENC /S2      S      64586
UTL1 label = UTL1 This is User Trailer Label 2
* Tape mark

-----
* Tape mark

-----
End of volume (double tape mark) reached, label summary follows
Label summary for volume DUTSUL:
Data Set Name      Blocks BLKSIZE LRECL RECFM Created Expires Security
1 DUTPLIB          622    3140  3136  VS 1990176          NO
2 DUTLOAD          122    6164  6160  VS 1990176          NO

```

Figure 46. A Tape Label Summary: Output from TLB

To print a summary (map) of the data contained on a tape, use the **TMP** function. You can produce a map of the entire tape, or of a specific number of tape files.

TMP prints from each file the number of blocks specified (in hexadecimal format or character format), followed by a summary of the space that the files take up on the tape. The output is affected by several **SET** parameters, as described in "Controlling Print Output" on page 79. Figure 47 on page 73 shows sample output from **TMP**.

Updating Tape Blocks in Full-Screen Mode

In full-screen mode, you can use the **TU** function to make interactive changes to tape blocks in a tape file, while copying it to another tape. (The data on the input tape remains unchanged.) At any time, both tapes are positioned at the same block within the file. When you end the TU function, the remaining data (up to the limit of files specified) is copied.

To update a tape block, perform the following steps:

1. Select the tape update function in either of these ways:
 - Select the function from the Tape Functions menu.
 - Enter TU on the command line of any DITTO panel.

The Tape Update panel appears. On this panel, you can use all the commands and function keys that you can use in browse functions (as described in "Browsing Data" on page 32), plus additional commands for updating tape blocks.

2. Select the first block that you want to update. Either move the cursor to the block, or use any of the locate commands that are available in browse functions.
3. Press the Zoom function key. This gives you a detailed view of the selected record.
4. Update the block in any of these ways:

- Overtyping the length field with a new record length.
- Overtyping the existing text with new text.
- Entering a column number in the Col field, and using the GET or COPY command to overwrite data in the block, starting at the specified column.

For the GET command, the data that you previously copied to the buffer with the PUT command is written. For the COPY command, you specify the columns of the current block that you want to copy.

For example, if the value in the Col field is 5 and you enter COPY 10-12, the data in columns 10 to 12 replaces the data that was previously in columns 5 to 7. If you enter COPY 10, the character in column 10 replaces the character that was previously in column 5.

5. Use the Exit function key to save your changes or the Cancel function key to discard the changes. A log of your changes is printed. You can select another block to update or leave the function.

The Tape Update Function: Special Considerations

There are several differences between the **TU** function and the other update functions (**DU**, **LU**, **OU**, and **VU**):

- **TU** copies tape blocks from one device to another device, with the changes that you specify. If you make a mistake, you still have the original data on the input tape as a backup.
- You can copy and update several consecutive tape files by specifying the number of files that you want to update. To update all the files until a double tape mark or an EOV label is reached, specify EOV.

- When you enter TU, both the input tape and the output tape are backspaced to the nearest tape mark (if they are not already positioned immediately after a tape mark).
- Blocks are copied dynamically, as you move through the files. For example:
 1. You enter the **TU** function and specify an input tape and an output tape. You accept the default (to update only one tape file).
One screen of data is displayed while copying to the output tape.
 2. You move forward 50 blocks.
The first 50 blocks are written to the output tape.
 3. You move backward 10 blocks.
Both tapes move back 10 blocks.
 4. You change the current block (block 41).
The changed block is written to the output tape.
 5. You end the TU function.
All blocks from block 42 to the end of the file are written to the output tape.
- Because blocks are copied dynamically, *always* make your updates in ascending order, starting at the beginning of the file. Simply scrolling backwards does not cause any updates to be lost. However, if you *update* a block (with a lower block number than previously changed blocks), all previous changes to blocks with a higher number are lost. Also, if you rewind the tapes, *all* updates are lost.
For example, if you change block 50, then change block 40, then end the TU function, this is what happens:
 1. Blocks 1 to 50 are copied.
 2. The changed block 50 is written.
 3. The changed block 40 is written.
 4. Blocks 41 to the end of the file are copied. The change that you made to block 50 is lost.
- Throughout the update process, the two tapes should remain in synchronization. If an input/output error causes the tapes to get out of synchronization, use the **REW** command from the command line to rewind the tapes and restore synchronization. (Note, however, that rewinding the tapes causes all previous updates to be lost.)

The **TU** function does not support ASCII conversion.

Updating One Physical Record at a Time

In line mode or full-screen mode, you can use the **TRL** function to change the contents of a physical tape record (block).

Like the **TU** function, **TRL** copies from one tape file to another tape file with the change that you specify. If you make a mistake, you still have the original data on the input tape as a backup.

Unlike the TU function, TRL lets you start and stop copying at any location within a file. You can therefore make a copy of a file with certain records removed. By using TRL repeatedly, you can combine data from two or more input files, or rearrange the data from a single input file.

To use TRL:

1. Position the input tape to the beginning of the tape file that contains the record you want to change. (If you want to copy a partial file, position the tape to the position where you want to start copying.)
2. Position the output tape to the position where you want to start copying.
3. Enter TRL, and specify the input and output tapes.
4. Move both tapes forward to the position of the first record that you want to update.

For example, if you want to update record 5, specify 5 when asked for a number of blocks and specify F (forward) when asked for a direction. Records 1 to 5 are copied to the output tape, and record 5 is displayed.

5. You are asked if this is the record that you want. If you say N (no), you can move the tapes forward or backward to another record.
6. You are asked if you want to change the length of the record. If you say Y (yes), you can change the record length.
7. You are prompted for the position within the record where your changes will begin.
8. You are prompted for the new text that will replace the existing text. You can enter the text in character form (surrounded by single quotation marks if it contains blanks, commas, or lowercase letters), or hexadecimal form.
9. You are then asked if the changes are complete.

If you respond Y, the block is copied to the output tape. A log of your changes is printed.

If you respond N, DITTO prompts you for additional changes to the same record.

10. There are two ways to end TRL:
 - Copy forward enough blocks to copy the rest of the file to the output tape.
 - Press the Exit function key or enter U in response to the Number of blocks prompt. TRL writes a tape mark on the output tape after the last block copied. The output tape is positioned immediately after the tape mark; the input tape is positioned immediately after the last block copied.

Figure 48 on page 77 shows a partial TRL session.

```

Number of blocks to reposition tapes for record load ?
4
Copy forward or backspace, F or B
f
Copying...
UHL1This is a Mark Shillington User Header Label
ECDFE88A48A484D8994E8899898A994EA894C888894D888944444
48313892092010419202893395736504259085145903125300000
Desired record ? Y or N
y
Changes to length ? Y or N
n
Starting data position in record
17
Shillington User Header Label
E8899898A994EA894C888894D88894444444444444444444444444444
289339573650425908514590312530000000000000000000000000
Enter new data (cc... or 'c...c' or X'nn...')
'Spannrng '
Spannrng User Header Label
E9899898444EA894C888894D8889444444444444444444444444444444
271559957000425908514590312530000000000000000000000000
Changes complete ? Y or N
y
Number of blocks to reposition tapes for record load ?
99
Copy forward or backspace, F or B
f
Copying...
4 block(s) and tape mark copied
DITTO function ?

```

Figure 48. TRL Dialog Example (Line Mode)

Viewing a Physical Tape Record at the System Console

In line mode, you can use the **TCN** function to view physical tape records (blocks) at the system console. (In full-screen mode, if you enter TCN, the TB function is automatically invoked.) Up to 50 bytes from the tape record or tape mark following the current tape position are displayed.

To invoke this function, enter TCN and identify the tape unit. Under MVS, you supply the name of a DD or TSO ALLOC statement, Under VSE or CMS, you supply a device address. Use SET ASCII=IN to translate data stored in ASCII format on tape for display.

Within TCN, you can use the following subcommands:

- D,n** Displays data in character format, starting with byte *n* if specified.
- DX,n** Displays data in hexadecimal updown format, starting with byte *n* if specified.
- P** Prints the current record in character format.
- PX** Prints the current record in hexadecimal updown format.

B Moves back to the previous record.

N Moves forward to the next record.

? Gives help information.

QUIT Ends the **TCN** function. The tape is positioned after the current record.

You can also use DITTO tape-positioning commands (**BSF**, **FSF**, **BSR**, **FSR**, and **REW**) within the **TCN** function.

If a data check occurs, an error message is displayed, and the part of the tape record that has been read (if any) is displayed.

Searching for Specific Data

In line mode, command mode, or batch mode, you can use the **TRS** function to find specific data on a tape. (In full-screen mode, you can also use **TRS**, but you will probably find **TB** more convenient.)

When using the **TRS** function:

- You identify the tape device.
- You can specify a record format and (if the records are fixed-length) the record length, to deblock the tape records while searching.
- You can specify where in each record you want to search. The default is to search for strings that start in column 1.
 - To search for strings that start in column *n*, reply **N0** to the pattern scan prompt and reply with the number *n* to the position prompt.
 - To search for strings anywhere in the record, reply **YES** to the pattern scan prompt.
 - To search for strings anywhere from column *n* to the end of the record, reply **YES** to the pattern scan prompt and reply *n* to the position prompt.
- You specify the string that you want to search for. The string can contain up to 50 characters and can be:
 - A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
 - A character string enclosed in single quotation marks. The string *can* contain blanks and commas; the string must match exactly (including case).
 - **X** followed by a hexadecimal string enclosed in single quotation marks, for example **X'C1C2C3'**.
- You can specify the number of occurrences of the string that should be found before the search stops. The default is to stop after the first occurrence.
- You can specify the number of records to search through. The default is to scan to the end of the file.

The **TRS** function stops after finding the specified number of occurrences of the string. The tape is positioned immediately after the record containing the last occurrence. If the string is not found, the tape stops after the next tape mark.

Printing Tape Files

Use the **TP** function to print tape records in character or hexadecimal format. You can specify the record format, the record length, the number of files that you want to print, and the maximum number of records to print per file.

The SET parameter `DUMP=UPDOWN` causes hexadecimal print requests to print in hexadecimal updown format; `DUMP=ACROSS` causes them to print in hexadecimal across format. See “Formats for Displaying or Printing Data” on page 4 for an example of each format.

Printing starts with the block following the current position and with the first block of each subsequent file. A leading tape mark is printed, if it exists. If you do not specify the number of blocks or files to be printed, one file is printed (up to the next tape mark).

Printing SYSLST Tape Files (VSE Only)

Use the **TFA** function to print SYSLST type A tape files (with CCW command codes), or use the **TFD** function to print SYSLST type D tape files (with ASA control characters). The corresponding functions for SYSLST disk files are **SFA** and **SFD**.

SYSLST type A files contain CCW command codes to control how each line is printed. The TFA function assumes that any line with a length of 121 characters was produced by job control, and therefore contains ASA control characters.

SYSLST type D files contain ASA control characters in the first character of each line. The TFD function translates ASA control characters into their corresponding CCW command codes, according to the table provided in *VSE/ESA System Macro User's Guide*. For any undefined character, a space and single line are printed.

You can specify the number of records that you want to print. The default is ALL.

Controlling Print Output

Several SET parameters affect the printed output:

| | |
|-----------------|--|
| DATAHDR | Whether the output contains a data header for each record |
| DBCSPRT | How double-byte characters are printed |
| DUMP | Which format (updown or across) is used for hexadecimal output |
| HEADERPG | Whether the output contains a header page at the beginning |
| PAGESIZE | How many lines per page the output has |
| PAGESKIP | Whether output from each function starts on a new page |
| PRINTLEN | How many columns wide the output is |
| PRINTOUT | Where the output is sent |
| PRTRANS | Whether non-printable characters are translated to blanks |
| RECLIMIT | Which part of each record to print |

For more information about SET parameter values, see “SET (Set Processing Options)” on page 242.

Copying Tape Records

The following DITTO functions let you copy tape records to different media:

- TV** Copies tape records to a VSAM data set. You can specify the number of records to skip from the beginning of the file and the number of records to copy.
- TS or TQ** MVS or VSE: copies tape records to a sequential data set. You can specify the number of records to skip from the beginning of the file and the number of records to copy.
- TF** CMS: copies tape records to a CMS file. You can specify the number of records to skip from the beginning of the file and the number of records to copy.
- TC** Copies one tape file to cards. For more information about card output, see “Card Input and Output” on page 144.
- TL** VSE: copies a tape file to a library.
- TX** Copies tape records to a REXX stem variable, if invoked from a REXX procedure. You can specify the number of records to skip from the beginning of the file and the number of records to copy. The **XT** function copies a REXX stem variable back to a tape file.

For the TF, TS, and TV functions, you can also specify the number of files that you want to copy, as described in “Copying Tape Data across a Network” on page 81.

Under MVS, QSAM functions can also process tapes with label processing.

Copying from Tape to Tape

The following DITTO functions let you copy from one tape to another tape:

- TT** Copies one or more files from tape to tape without label processing. Tape labels are treated as data. Thus, if multiple output volumes are required, the label information may be incorrect.
- TTR** Copies one file from tape to tape. You can change the record format and reblock the tape file. Tape labels are treated as data.
- TFT** VSE or CMS: copies one file, or part of one file, from a standard labeled tape or an unlabeled tape to a standard labeled tape. You can change the record format and reblock the tape file.
- TLT** Copies IBM-standard labeled tapes (without requiring you to specify every file ID). Labels read from the input tape are interpreted and copied to the output tape. Copying multivolume and multifile tapes is supported. However, multivolume output tapes are not supported: if the output volume is too short, copying continues on the next tape, but no end-of-volume (EOV) label is written on the full output volume, and no IBM-standard VOL1, HDR1, and HDR2 labels are written on the continuation volume.

You can copy all of the files on the tape (by default) or specify the number of files that you want to copy. You can start copying at a specific file by positioning the tape at the header label (HDR1) of the desired file. You can also append files on the output volume by positioning the output tape after the last file (EOF2 TM) of the output volume.

You can copy a tape that contains both labeled and unlabeled files by specifying that the type is MIXED. Non-labeled files are tolerated and copied as non-labeled files.

Note: Any user labels on the input tape are not copied.

Additional Notes for MVS Only:

1. DITTO may force bypass label processing (BLP) when switching to the next volume. If BLP is forced, for security reasons the tape is freed (unallocated) when TLT ends.
2. HSM backup tapes should not be copied using **TLT** because DITTO cannot update the HSM internal information regarding the location of the backup data sets.
3. No recataloging or RACF processing of the data sets takes place because of data set naming conventions on tape.
4. In some error conditions (for example, if unexpired output volumes are used), prompts are sent to the operator console.

Copying Tape Data across a Network

To send the contents of a tape across a network:

1. Use the **TF**, **TS**, or **TV** function (with the *nfiles* parameter) to copy all the files on the tape, including tape marks, to a single CMS file, sequential data set, or VSAM entry-sequenced data set.

Each tape mark is converted to an 80-character record that has the format:

```
DITTOTMDITTOTMDITTOTMDITTOTMDITTOTMDITTOTMDITTOTMDITTOTMDITTOTMDIT
```

2. Transfer the file or data set across the network, using a file transfer program, such as the IBM File Transfer Program.
3. At the other end, use the **FT**, **ST**, or **VT** function to copy the file or data set to a tape. DITTO converts the tape-mark records back to tape marks on the output tape.

ASCII Translation

For all tape input and output functions except INT, TB, TFA, TFD, TLB, TLT, TTC, and TU, you can use the **SET** parameter ASCII to translate tape data between ASCII and EBCDIC format. For the TLB function, ASCII translation is automatically performed when needed. For the INT and TB functions, you can specify ASCII input or output when invoking the function. For the other functions, translation is not supported by DITTO.

ASCII data is translated to EBCDIC format using the DITA2E table. EBCDIC data is translated to ASCII format using the DITE2A table. To translate data from one ASCII format to another ASCII format, customize these table so that DITA2E is different from DITE2A. (For details, see the *DITTO/ESA Installation and Customization Guide*.)

To translate tape data, select the appropriate value of the **SET** parameter ASCII:

- To translate ASCII tape data to EBCDIC format, specify ASCII=IN
- To translate EBCDIC data to ASCII format for tape output, specify ASCII=OUT
- To translate tape data from one ASCII format to another, specify ASCII=BOTH
- To not translate data, specify ASCII=N0 (the default).

Note: Translating labels or varying-length format tapes may produce incorrect output.

Comparing Tapes

You can use the **TTC** function to compare, byte by byte, the data on one tape with the data on another. This function is particularly useful if you have made a copy of a tape, and want to check that the duplicate is exactly the same as the original.

You can compare an entire tape, or a specific number of files. The compare function stops processing as soon as it finds the first difference. Be sure to position the tapes as needed before starting the comparison.

To compare all the files on the tapes, specify the value EOV for the parameter *nfiles*. If you do not specify a value for *nfiles*, only one file (from the current block to the next tape mark) is compared.

When a difference is found, DITTO lists:

- The two records that do not match
- The lengths of these records on the two tapes
- The location of the first different byte
- The number of blocks and bytes compared

If no difference is found, DITTO lists the number of blocks and bytes compared.

Note: An input/output error is considered a difference, even if the same error occurs on both tapes.

Initializing a Tape

Use the **INT** function to initialize a tape. The tape is rewound, and the appropriate header information is written.

- To initialize an IBM standard labeled tape, supply the volser, specify N0 for the ASCII field, and specify TAPELBL=SL in the SET parameters. You can also supply a file ID for the header label.
- To initialize a tape for ASCII files (with ANSI labels), supply the volser, specify YES for the ASCII field, and specify TAPELBL=AL or TAPELBL=AL4 in the SET parameters. You can also supply a file ID for the header label.
- To initialize a no label (NL) tape, do not supply a volser.
- You can use the SCALE field to limit the amount of data that can be stored in the tape to a percentage of its capacity. Allowed values are 0 (100% capacity; no limit applied), 25, 50, or 75. For example, a value of 25 limits the tape to 25% percent of its normal capacity. This field is valid only for 3590 devices; it is not valid for a 3590 running in 3489E emulation mode.

After completion of the function, the tape remains positioned after the label set.

Under MVS, it is recommended that you unload the tape and reallocate it with the new volume serial number, to make your system aware of the new volume serial number.

Figure 49 shows a sample batch job for initializing a tape under MVS.

```

//*****
//* SAMPLE JOB TO INITIALIZE A TAPE WITH DITTO/ESA UNDER MVS
//*****
//*
//DTOBAT   PROC
//DTOBAT   EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//         PEND
//*
//INITTAPE EXEC DTOBAT
//TAPE     DD DISP=(SHR,KEEP),DSN=DUMMY,LABEL=(,BLP),UNIT=CART,
//         VOL=(,RETAIN,SER=DIT001)
//SYSIN DD *
$$DITTO REW LABEL=BLP,OUTPUT=TAPE
$$DITTO INT OUTPUT=TAPE,VOLSER=DIT001
$$DITTO RUN OUTPUT=TAPE
/*

```

Figure 49. Batch Example of Tape Initialization (MVS)

Creating Tape Records

Use the **BT** function to write simple test records to a tape file. You specify the tape that will contain the records, the number of records, and the record length. You can also specify a fill character or fill pattern, a sequence field, a record format, and a block size.

Erasing Tape Records

The **ERT** function:

1. Writes two tape marks at the current tape position
2. Erases the remaining tape data to the end of the tape (data security erase)
3. Rewinds the tape

ERT can be used only with tape units that support the data security erase I/O command code, such as the IBM 3400 series.

Examples of DITTO Tape Functions

Figure 50, Figure 51 on page 85, and Figure 52 on page 85 show an MVS batch job, a VSE batch job, and a CMS REXX procedure that work with tapes. In each example, a tape is copied and then part of the output tape is printed.

- A** MVS: DD statements for the input and output tape. Because label processing is bypassed (BLP), the DSN specified can be DUMMY.

VSE: an assignment statement for the input tape. Because no assignment statement is provided for the output tape, any DITTO commands that refer to the output tape must include its physical address.
- B** DITTO control statements to rewind the input and output tape to the load point (begin-of-tape, BOT).
- C** DITTO control statement for tape-to-tape copy. The first 14 files of the input tape are written to the output tape, using the DITTO function **TT**. Output is written to the output tape starting at the current position (the load point in this case).
- D** DITTO control statement for tape printing. The first two files of the output tape are printed, in character format, using the DITTO function **TP**. If the limit of 10 records per file is reached in either the first file or the second file, printing stops.

```

//S1      EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
A //INTAPE DD DISP=(OLD,KEEP),DSN=DUMMY,LABEL=(,BLP),
//          UNIT=480,VOL=SER=DUTTP1
A //OUTTAPE DD DISP=(NEW,KEEP),DSN=DUMMY,LABEL=(,BLP),
//          UNIT=380,VOL=SER=DUTTP2
//SYSIN   DD *
B $$DITTO REW INPUT=INTAPE
B $$DITTO REW OUTPUT=OUTTAPE
C $$DITTO TT INPUT=INTAPE,OUTPUT=OUTTAPE,NFILES=14
    $$DITTO REW INPUT=OUTTAPE
D $$DITTO TP INPUT=OUTTAPE,NFILES=2,NLRECS=10
    $$DITTO EOJ
/*
//

```

Figure 50. Batch Example Using DITTO Tape Functions (MVS)

```

// JOB TT COPY TAPE TO TAPE
// UPSI 1
A // ASSGN SYS012,480,C0
// EXEC DITTO
B $$DITTO REW INPUT=SYS012
B $$DITTO REW OUTPUT=481
C $$DITTO TT INPUT=SYS012,OUTPUT=481,NFILES=14
    $$DITTO REW OUTPUT=481
D $$DITTO TP INPUT=481,NFILES=2,NLRECS=10
    $$DITTO EOJ
/&

```

Figure 51. Batch Example Using DITTO Tape Functions (VSE)

```

/* REXX */
/* Example: Execute Tape Copy and Print Commands */
B 'DITTO $REW INPUT=480'
B 'DITTO $REW OUTPUT=481'
C 'DITTO $TT INPUT=480,OUTPUT=481,NFILES=14'
    'DITTO $REW OUTPUT=481'
D 'DITTO $TP INPUT=481,NFILES=2,NLRECS=10'

```

Figure 52. Command Mode Example Using DITTO Tape Functions (CMS)

Tape

Chapter 7. Processing Physical Disk Records

You can use the following functions to process disk records:

Figure 53. DITTO Functions for Disk Processing

| Purpose | Functions | |
|---|------------|--------------------|
| Get VTOC information for disk volumes, sorted by data set name, extent, creation date, or expiration date | DVT | Display VTOC |
| Rename or delete VTOC entries | PVT | Process VTOC |
| Alter a disk volume identification | DID | Alter Disk VOLID |
| Display the extents of a disk data set | DSE | Data Set Extent |
| Browse disk records at your terminal | DB | Disk Browse |
| View a single disk record at the system console | DCN | Disk to Console |
| Update disk records at your terminal | DTE | Disk Track Edit |
| | DRL | Disk Record Load |
| | DU | Disk Update |
| Print disk records in character or hexadecimal format | DP | Disk Print |
| Locate data on a disk | DRS | Disk Record Scan |
| Write a disk end-of-file (null) record | EOF | End-of-File Record |

Processing Disk Records

Under MVS, you can process physical disk extents *within a data set*. You may also be able to use disk fullpack processing, as described in “MVS: Using Disk Fullpack Processing” on page 89. DITTO supports all count-key-data (CKD) disk storage devices that are supported by MVS.

Under MVS, each disk function works with a single volume. If a data set is split across two or more volumes, disk functions apply to the volser that you specify or (if the data set is cataloged) the first volume listed for the data set in the catalog.

Under VSE or CMS, you can process physical disk extents. DITTO supports all count-key-data (CKD) and fixed-block-architecture (FBA) disk storage devices that are supported by VSE and CMS.

Specifying Disk Input and Output

MVS:

- All disk functions normally work with a disk data set. For some functions, you may be able to work with an entire disk volume as described in “MVS: Using Disk Fullpack Processing” on page 89.
- To work with a disk data set, specify the data set name. If the data set is not cataloged, specify the volser of the disk.
- To work with an entire disk volume (if you are authorized), specify the volser of the disk and do not specify a data set name.

VSE and CMS:

- All disk functions work with an entire disk volume.
- To work with a disk volume, specify the unit address (3 or 4 digit physical device address) or the letter V followed by the volume serial number of the disk. (In full-screen mode, the unit and volser are specified in different fields, so you do not need to specify V before the volser.)

MVS: Specifying Disk Extents

In most disk functions, you can specify a disk address or a disk address range. You can specify these addresses in the form *cylhd*, where the last 2 digits represent the head value and the other digits represent the cylinder.

Under MVS, if the SET parameter CYLHD has the value ABSOLUTE, each *cylhd* value represents an absolute address (counting from the beginning of the physical disk). If the SET parameter CYLHD has the value RELATIVE, each *cylhd* value represents an address relative to the beginning of the data set. (Under VSE or CMS, disk addresses are always absolute.)

In full-screen mode, the entry panels for disk functions tell you whether absolute addresses or relative addresses are in effect.

In line mode, the prompts for addresses tell you whether absolute addresses or relative addresses are in effect.

In command mode or batch mode, to be sure of getting the desired records you should execute a SET CYLHD command before executing any disk commands.

MVS: Using Disk Fullpack Processing

Under MVS, you may have access to disk fullpack processing. For information about when disk fullpack processing is available, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

With disk fullpack processing, you can browse, update, or print data on an entire DASD volume. To invoke a disk function with fullpack processing:

1. Invoke the function (for example, **DB**).
2. Do not specify a value in the data set name field (or for the data set name prompt).
3. Specify a value for the volser field (or prompt).

For example, to invoke the DP function with disk fullpack processing in command mode, you could enter:

```
DITTO $DP,VOLSER=VOL001,BEGIN=R0,END=N200
```

Listing the Files on a Disk

You can use the **DVT** (Display VTOC) function to display or print a list of data sets on a disk. This function can be used only for OS (MVS) or DOS (VSE) formatted disks. You can list either all the data sets on the disk, or a subset of the data sets based on a generic data set name.

The information in the list includes:

- Data set name
- Disk extents
- File organization information
- Creation and expiration dates

You can sort the list by data set name, creation date, expiration date, or disk extents.

Note: DVT shows “never to expire” expiration dates of 99/999 and 99/366 as 9999.999 and 9999.366, respectively. An expiration date that was explicitly specified as 99/365 is also considered “never to expire,” and is shown as 9999.365. However, an expiration date of 99/365 that was calculated from a specified retention period (current date + number of days = 99/365) *will* expire, and is shown as 1999.365. If you sort the list by expiration date, values beginning with 9999 appear first.

In line mode or batch mode, you can use DVT to print a list of files. In full-screen mode, you can also use DVT to display a full-screen file list; from this list, you can select the file that you want to work with. In command mode under MVS or CMS, entering DVT invokes the full-screen mode DVT function.

To use DVT under CMS, you must have CMS/DOS.

Using the DVT Function in Full-Screen Mode

1. Select the DVT option from the Disk Functions menu, or enter DVT on the command line of any panel in DITTO.
2. On the panel that appears, you specify the disk that you want to work with. Specify the volume serial number or (VSE and CMS only) the physical device address.

You can also specify a generic data set name (to limit the data sets that are listed), a sort order, and whether you want the list to be displayed or printed.

3. DITTO lists the files.

```

Process  View  Options  Help
-----
DITTO/ESA for VSE          DVT - Display VTOC                      Line 1 of 41

Unit 0241 VOLSER SYSWK1 3390 with 1113 cyls, 15 trks/cyl, 58786 bytes/trk

--- Data Set Name --- sorted by NAME  ----- Ext      Begin-end      Reltrk,
1...5...10...15...20...25...30...35...40.... seq  Cyl-hd  Cyl-hd      numtrks
*** VTOC EXTENT ***
CCCA.CICS.DUMPA           0      60  2    62  1    902,30
CCCA.CICS.DUMPB          0      62  2    64  1    932,30
DITTO.TEST.NLS.LIBRARY   0     1097  6   1098  9   16461,19
DOS.LABEL.FILE.FF0000D99672.AREA1 0     1111  0   1112  14  16665,30
DOS.PAGING.FILE.FF0000D99672      0      850  0   1046  14  12750,2955
FQOC411.FILE01           1     1100  0   1100  9   16500,10
HISTORY.FILE.INSTALL.MLX      0      575  0    575  14   8625,15
HISTORY.FILE.LOAD.MLX        0      573  0    573  14   8595,15
ICCF.LIBRARY              0      302  0    421  14   4530,1800
INFO.ANALYSIS.DUMP.MGNT.FILE      0      570  0    570  14   8550,15
INFO.ANALYSIS.EXT.RTNS.FILE      0      571  0    571  14   8565,15

Command ==> _____ Scroll PAGE
F1=Help  F2=Browse  F3=Exit  F4=Process  F5=Rgtleft  F6=Rfind  F7=Bkwd  F8=Fwd
F10=Actions  F11=CRetrieve  F12=Cancel

```

Figure 54. Display VTOC

By using the Rgtleft function key you can switch between two display formats to see all the information you would get on a full-width printout. The second display format is shown in Figure 55 on page 91.


```

Process  View  Options  Help
-----
DITTO/ESA for VSE          DVT - Display VTOC          Line 1 of 41
Unit 0241 VOLSER SYSWK1 3390 with 1113 cyls, 15 trks/cyl, 58786 bytes/trk

--- Data Set Name --- sorted by NAME  --  RECFM  BLKSIZE  Created  Expires
1...5...10...15...20...25...30...35...40. DSORG  LRECL          YYYY.DDD  YYYY.DDD
*** VTOC EXTENT ***
CCCA.CICS.DUMPA             SAM      0      0 1999.089 1999.089
CCCA.CICS.DUMPB             SAM      0      0 1999.096 1999.096
DITTO.TEST.NLS.LIBRARY      SAM      0      0 1999.070 1999.070
DOS.LABEL.FILE.FF0000D99672.AREA1  UND     0      0 1999.082 9999.366
DOS.PAGING.FILE.FF0000D99672      UND     0      0 1999.089 9999.366
FQOC411.FILE01             SAM      0      0 1998.106 1998.106
HISTORY.FILE.INSTALL.MLX      SAM      0      0 1998.140 1998.140
HISTORY.FILE.LOAD.MLX        SAM      0      0 1998.140 1998.140
ICCF.LIBRARY                DAM      0      0 1998.206 9999.366
INFO.ANALYSIS.DUMP.MGNT.FILE  SAM      0      0 1999.089 1999.089
INFO.ANALYSIS.EXT.RTNS.FILE  SAM      0      0 1999.089 9999.365

Command ==> _____ Scroll PAGE
F1=Help  F2=Browse  F3=Exit  F4=Process  F5=Rgtright  F6=Rfind  F7=Bkwd  F8=Fwd
F10=Actions  F11=CRetrieve  F12=Cancel

```

Figure 55. Display VTOC (Alternative Display)

To move forward or backward through the list, use the Fwd or Bkwd function key, or use the locate command. You can also enter a value in the Scroll field. (In DITTO, all full-screen list functions and full-screen browse functions use the same basic methods for locating data. These methods are described in Chapter 3, “Browsing Data in Full-Screen Mode” on page 31.)

To work with a data set, move the cursor to the data set name, and press the Process function key. (If the cursor is outside the list area, you will work with the first entry in the list.) The selected line is highlighted in another color. A pull-down menu appears below the action bar. Select an option on the pull-down menu.

You can also type the following line commands over the beginning of the file ID:

browse Invokes DB
directory VSE: invokes LDL
edit Invokes DTE
view Invokes APB
extents Invokes DSE
print Invokes DP
rename Invokes PVT with the RENAME option, under VSE or CMS
scratch Invokes PVT with the SCRATCH option, under VSE or CMS
status VSE: invokes LSL
update Invokes DU

Under VSE or CMS, you can also use the **PVT** function to rename or delete entries in the disk VTOC. To use PVT under CMS, you must have CMS/DOS.

VSE: Displaying the Contents of the Label Area

Use the **DLA** function to display or print the labels stored in the label area of your VSE/ESA system. You can select all labels, specify a label subarea, or specify a generic name. You can exclude temporary labels.

The output contains one line for each DLBL (TLBL) statement or DLBL/EXTENT pair, and additional lines for follow-on extent statements. It covers all DLBL, EXTENT, and TLBL parameters.

On a full-screen display, you can toggle between the primary and alternate displays and use the MORE command to show additional label parameters that are not included in the default output.

You can use the full-screen display as a selection list for various DITTO functions that work with libraries, VSAM catalogs and other data sets. Press the Process function key and select the desired function from the pull-down menu, or type a line command over the beginning of the label name.

For more information about the DLA function, see the online help.

Locating a Disk Data Set

Use the **DSE** function to see the disk extents of a particular data set. This function can be used only for OS (MVS) or DOS (VSE) formatted disks. Figure 56 shows an example of DSE output for MVS.

CMS: CMS/DOS is required.

```

Process  View  Options  Help
-----
DITTO/ESA for MVS      DSE - Disk Data Set Extents

DITTO/ESA Messages
VOLSER: TS0141  Data Set Name: MARK9462.MULTI.EXTENT
More: +

Absolute and relative extent limits within data set:

          Begin - ABSOLUTE - End          Begin - RELATIVE - End
Ext  Tracks  Cyl-hd Rnnnnn  Cyl-hd Rnnnnn  Cyl-hd Rnnnnn  Cyl-hd Rnnnnn
0    5  186  9  2799  186 13  2803    0  0    0    0  4    4
1    5  291 12  4377  292  1  4381    0  5    5    0  9    9
2    5   69  0  1035   69  4  1039    0 10   10   0 14   14
3    5   69  5  1040   69  9  1044    1  0   15    1  4   19
4    5   69 10  1045   69 14  1049    1  5   20    1  9   24
5    5   70  0  1050   70  4  1054    1 10   25    1 14   29
6    5  363  4  5449  363  8  5453    2  0   30    2  4   34

Press the CANCEL or EXIT key to remove the message Pop-Up
Command ==>
F1=Help  F3=Exit  F7=Bkwd  F8=Fwd  F12=Cancel

F1=Help  F3=Exit  F10=Actions  F11=CRetrieve  F12=Cancel

```

Figure 56. Data Set Extents Display (MVS)

Browsing and Updating Disk Records

In full-screen mode, you can use the **DB** function to browse disk records. For more information about this function, see Chapter 3, “Browsing Data in Full-Screen Mode” on page 31.

Updating Disk Records in Full-Screen Mode

If you want to edit a disk track in full-screen mode, you can use the **DTE** function, as described in Chapter 4, “Editing Data in Full-Screen Mode” on page 39. You can also use the **DU** function (although DTE is usually more convenient).

If you decide to use the DU function, perform the following steps:

1. Select the disk update function in one of these ways:
 - Select the function from the Disk Functions menu.
 - Enter DU on the command line of any DITTO panel.
 - Select the Update function from the Process pull-down menu in the DVT list (or the SCS list under MVS).
 - Type update over the *beginning* of the file ID in the DVT or SCS list.

The Disk Update panel appears. On this panel, you can use all the commands and function keys that you can use in browse functions (as described in “Browsing Data” on page 32), plus additional commands for updating records.

2. Select a record that you want to update. Either move the cursor to the record, or use any of the locate commands that are available in browse functions.
3. Press the Zoom function key. This gives you a detailed view of the selected record.
4. Update the record in any of these ways:
 - Overtyping the existing text with new text.
 - Entering a column number in the Col field, and using the GET or COPY command to overwrite data in the record, starting at the specified column.

For the GET command, the data that you previously copied to the buffer with the PUT command is written. For the COPY command, you specify the columns of the current record that you want to copy.

For example, if the value in the Col field is 5 and you enter COPY 10-12, the data in columns 10 to 12 replaces the data that was previously in columns 5 to 7. If you enter COPY 10, the character in column 10 replaces the character that was previously in column 5.

5. Use the Exit function key to save your changes or the Cancel function key to discard the changes. A log of your changes is printed. You can select another record to update or leave the function.

Updating One Record at a Time

In line mode, you can use the **DRL** function to change the contents of a specific disk record. (In full-screen mode, you can also use DRL, but you will probably find DU more convenient.)

- MVS: enter the name of the data set, the volume serial number, or both.
- VSE or CMS: enter the physical device address of the disk or V followed by the volume serial number of the disk.

- Enter the location of the record within the data set.
MVS: the location is either absolute or relative, as described in “MVS: Specifying Disk Extents” on page 88.
VSE or CMS: for an FBA disk, you can also enter the CI size.
- DITTO displays the record. If this is not the record that you want to change, exit the function and invoke it again.
- DITTO prompts you for information about the character string you want to change.
- After each change, you are asked whether the changes are complete. If you reply Y, the record is written to disk. A log of your changes is printed.

Viewing a Disk Record at the System Console

In line mode you can use the **DCN** function to view disk records at the system console. (In full-screen mode, if you enter DCN, the DB function is automatically invoked.) Up to 50 bytes from the disk record are displayed.

To invoke this function:

1. Enter DCN.
2. MVS: enter the name of the data set, the volume serial number, or both.
VSE or CMS: enter the physical device address of the disk or V followed by the volume serial number of the disk.
3. Enter the location of the record.
MVS: the location is either absolute or relative, as described in “MVS: Specifying Disk Extents” on page 88.
VSE or CMS: for an FBA disk, you can also enter the CI size.

Within DCN, you can use the following subcommands:

| | |
|--------------------|---|
| CNT | Displays the RECORD COUNT field data for a CKD device |
| D,<i>n</i> | Displays data in character format, starting with byte <i>n</i> if specified |
| DX,<i>n</i> | Displays data in hexadecimal updown format, starting with byte <i>n</i> if specified |
| K,<i>n</i> | Displays the CKD key in character format, starting with byte <i>n</i> if specified |
| KX,<i>n</i> | Displays the CKD key in hexadecimal updown format, starting with byte <i>n</i> if specified |
| P | Prints the current record in character format |
| PX | Prints the current record in hexadecimal updown format |
| B | Displays the previous record ('B' represents 'back') |
| N | Displays the next record |
| <i>location</i> | Displays the record at this address |
| ? | Gives help information |
| QUIT | Ends the DCN function |

Printing Disk Records

Use the **DP** function to print the records in a disk extent in character or hexadecimal format. The SET parameter **DUMP=UPDOWN** causes hexadecimal print requests to print in hexadecimal updown format; **DUMP=ACROSS** causes them to print in hexadecimal across format. See “Formats for Displaying or Printing Data” on page 4 for an example of each format.

Specifying What You Want to Print

You must specify which disk records you want to print. How you do this depends on whether the disk is a CKD disk or an FBA disk.

Specifying a Range for a CKD Disk

To print records from a CKD disk, you must specify the range of tracks that you want to print.

1. Specify the beginning of the range, in one of the following ways:
 - The beginning address in the form *cylhd*, where the last two digits represent the head and the other digits represent the cylinder.
 - A track number, in the form *Rnnnnnn*
 - Under MVS, an asterisk (*), which means the beginning of the data set

MVS: Depending on the value of the SET parameter **CYLHD**, you specify either the absolute disk location or the relative location within the data set.

2. Specify the end of the range, in one of the following ways:
 - The end address in the form *cylhd*
 - A track number, in the form *Rnnnnnn*
 - A number of tracks, in the form *Nnnnnnn*
 - Under MVS, an asterisk (*), which means the end of the data set

MVS: Depending on the value of the SET parameter **CYLHD**, you specify either the absolute disk location or the relative location within the data set.

If a defective track with an assigned alternate track is encountered, **DITTO** processes the contents of the alternate track.

When you print records from a CKD disk, you can specify any of the following dump types:

- KD** Prints the key fields (if any) and data fields. This is the default.
- CKD** Prints the home address and record 0 fields, followed by the count, key, and data fields
- KEY** Prints *only* the key field, and not the data portion of each record
- CNT** Prints *only* the home address and count fields, and not the key field and data portion of each record

Specifying a Range for an FBA Disk

To print records from an FBA disk, you must specify the range of physical or logical records that you want to print:

1. Specify the physical block number (PBN) of the beginning of the range.
2. Specify the end of the range, in either of the following ways:
 - A physical block number, in the form *nnnnnnn*
 - A number of blocks, in the form *Nnnnnnnn*

When you print records from an FBA disk, you process physical records by default. To process logical records, specify a control interval size greater than 0. If you are processing logical records, the first block number that you specify must be the first block in a control interval. DITTO processes only control intervals that fit entirely within the block range that you specify.

Controlling Print Output

Several SET parameters affect the printed output:

| | |
|-----------------|--|
| DATAHDR | Whether the output contains a data header for each record |
| DBCSPRT | How double-byte characters are printed |
| DUMP | Which format (updown or across) is used for hexadecimal output |
| HEADERPG | Whether the output contains a header page at the beginning |
| PAGESIZE | How many lines per page the output has |
| PAGESKIP | Whether output from each function starts on a new page |
| PRINTLEN | How many columns wide the output is |
| PRINTOUT | Where the output is sent |
| PRTRANS | Whether non-printable characters are translated to blanks |
| RECLIMIT | Which part of each record to print |

For more information about SET parameter values, see “SET (Set Processing Options)” on page 242.

You can also specify a record length, to deblock each physical record into logical records. The physical block size does not need to be a multiple of the record length that you specify.

Writing a Disk End-of-File Record

The **EOF** function retrieves a disk record and rewrites it as a software end-of-file (EOF) indicator. For a CKD disk, the EOF record has a data length of zero; for an FBA disk, a control interval consisting entirely of zeros is written.

If a disk-write operation ends abnormally without writing an end-of-file record, you might want to use the EOF function to correct the problem.

This function is not available in batch or command mode.

Displaying or Changing a Disk Volume ID under VSE or CMS

Under VSE or CMS, the **DID** function displays or changes the volume ID of a disk. To display the volume ID, specify **DID** followed by either the physical device address of the disk or **V** followed by the volume serial number of the disk. To change the volume ID, also specify the new volume ID.

Searching for Specific Data

In line mode, command mode, or batch mode, you can use the **DRS** function to find specific data on a disk. (In full-screen mode, you can use either the **DB** function or the **DRS** function.)

When using the **DRS** function:

- You can limit the search to a range of records. (You select which records to search in the same way that you select which records to print in a print function. For more information, see “Specifying What You Want to Print” on page 95.)
 - You specify the type of scan that you want to perform:
 - D** Searches the data field of each record, *only at* position 1 (or the position that you specify)
 - DP** Searches the data field of each record, everywhere in the field (or *starting at* the position that you specify)
 - K** Searches the key field of each record, *only at* position 1 (or the position that you specify)
 - KP** Searches the key field of each record, everywhere in the field (or *starting at* the position that you specify)
 - E** Searches for end-of-file records. If you select **E**, you do not need to specify a search string.
- Note:** **K** and **KP** apply to CKD disks only.
- You can specify a record length, to deblock the physical records into logical records before searching.
 - You can specify a position at which the scan will begin (the default is 1).
 - You specify the string that you want to search for. The string can contain up to 50 characters and can be:
 - A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.
 - A character string enclosed in single quotation marks. The string *can* contain blanks and commas; the string must match exactly (including case).
 - **X** followed by a hexadecimal string enclosed in single quotation marks, for example, **X'C1C2C3'**.

The string does not apply if the scan type is **E**. Otherwise, it is required.

The output of the **DRS** function is a list of records that meet your search conditions. Figure 57 on page 98 and Figure 58 on page 98 show how to use **DRS** under MVS and under VSE.

```

DITTO function?
drs
Data set name ?
'keistew.mvs.jcl'
Begin, relative to data set - enter (c)ccchh or 'R'n...n track
r0
End - enter (c)ccchh or 'N'n...n tracks
n2
Scan key - K, data - D, EOF - E, (KP or DP for pattern scan)
d
Starting data position in record

Enter argument (CC... or 'C..C' or X'NN...')
tstfile1
Records at
cccc-hh-rrr - Position
295-00-017 - 00001

```

Figure 57. Searching for Disk Data with DRS (MVS)

```

DITTO function?
drs
Input disk ? (Device number or 'V'xxxxxx VOLID)
1ff
Begin, - enter c...chh or 'R'n...n track
280
End, - enter c...chh, 'R'n...n track, or 'N'n...n number of tracks
n200
Scan key - K, data - D, EOF - E, (KP or DP for pattern scan)
d
Starting scan position in record

Enter argument (cc... or 'c...c' or X'nn...')
tstfile1
Records at
cccc-hh-rrr - Position
295-00-017 - 00001

```

Figure 58. Searching for Disk Data with DRS (VSE)

Command Mode Example Using DITTO Disk Functions

Figure 59 on page 99 shows a REXX procedure that prints the first record of a disk data set.

- A** DITTO invocation so that the cylinder-head and track numbers specified in disk functions are interpreted as relative values (that is, relative to the beginning of the data set) and hexadecimal output is printed in across format.
- B** DITTO invocation to print a disk record in hexadecimal format. For the first track of the data set, the home address and all count, key, and data fields are printed.


```

/* REXX */
/* Example: Execute a Disk Print Command from a REXX procedure */
A 'DITTO $SET CYLHD=RELATIVE,DUMP=ACROSS'
B 'DITTO $DP DSN=TYRONED.V.VAR,BEGIN=R0,END=N1,FORMAT=HEX,LISTONLY=CKD'

```

Figure 59. Command Mode Example Using DITTO Disk Functions (MVS)

Batch Examples Using DITTO Disk Functions

Figure 60 shows a DITTO batch job under MVS that prints the first record of a disk data set.

- A** DD statement for disk input.
- B** DITTO control statement so that the cylinder-head and track numbers specified in disk functions are interpreted as relative values (that is, relative to the beginning of the data set), and hexadecimal output is printed in across format.
- C** DITTO control statement to print a disk record in hexadecimal format. For the first track of the data set, the home address and all count, key, and data fields are printed.

```

//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//S1 EXEC DTOBAT
A //DISKIN2 DD DSN=KEISTEW.DITTO.CNTL,DISP=SHR
//SYSIN DD *
B $$DITTO SET CYLHD=RELATIVE,DUMP=ACROSS
C $$DITTO DP INPUT=DISKIN2,BEGIN=R0,END=N1,FORMAT=HEX,LISTONLY=CKD
$$DITTO EOJ
/*

```

Figure 60. Batch Example Using DITTO Disk Functions (MVS)

Figure 61 shows a DITTO batch job under VSE that prints the first 100 records from a CKD disk with volser XYZ012.

```

// JOB DP PRINT FIRST 100 DISK RECORDS
// UPSI 1
// EXEC DITTO
$$DITTO DP INPUT=VXYZ012,BEGIN=R0,END=N100
$$DITTO EOJ
/&

```

Figure 61. Batch Example Using DITTO Disk Functions (VSE)

Disk

Chapter 8. Processing Sequential Data (SAM and QSAM)

You can use the following functions to process sequential data in QSAM (MVS only) or SAM-managed data sets:

Figure 62. DITTO Functions for Processing Sequential Data

| Purpose | Functions | |
|---|---|---|
| Print sequential data in character or dump format | SP, QP | Sequential Data Print |
| Print SYSLST disk files | SFA SFD | Print SYSLST File Type A Print SYSLST File Type D |
| Copy sequential data to cards, tape, CMS file, library member, OAM object collection, sequential data set, or VSAM data set | SC, QC ST, QT SF SL SO, QO SS, QQ SV, QV | Sequential Data to Card Sequential Data to Tape SAM to CMS File Sequential Data to Library Member Sequential Data to Object Sequential to Sequential Sequential to VSAM |
| Create records in a sequential data set | BS, BQ | Create Sequential Data |

Specifying Sequential Input and Output

Functions that use sequential input or produce sequential output are available in all modes.

MVS: various functions use QSAM to process logical records in sequential data sets. DITTO supports sequential data sets on tape, disk, or SYSIN/SYSOUT data. DITTO also supports members of partitioned data sets. You can define sequential input and output with JCL or without JCL.

VSE: various functions use SAM input or produce SAM output. You can also use VSAM-managed SAM files as input or output if you have the VSE/VSAM Space Management for SAM feature. You can define sequential input and output with JCL or without JCL.

CMS: functions that produce SAM output are not available. You do not use JCL to define SAM input. To work with SAM input files, you must have CMS/DOS.

Notes:

1. Sequential input or output must be completely defined with JCL or completely defined through DITTO. You cannot mix JCL statements and DITTO keywords when defining input or output.
2. Not all combinations of DITTO keyword values are supported by the underlying operating environment. For information on supported combinations, see your operating environment documentation.

MVS: Specifying Sequential Input and Output with JCL

Under MVS, you can use job control language (JCL) to define sequential input and output, as follows:

- Specify a DD JCL or TSO ALLOC statement for the data set. For an output data set, supply DCB information. For interactive DITTO, you must use QSAMIN for an input data set or QSAMOUT for an output data set.
- On the DITTO command, specify INPUT= or OUTPUT= followed by the ddname. If you used the default ddname (QSAMIN for QSAM input or QSAMOUT for QSAM output), you do not need an INPUT or OUTPUT parameter.

VSE: Specifying Sequential Input and Output with JCL

Under VSE, you can use job control language (JCL) to define sequential input and output, as follows:

- In the JCL, specify a DLBL statement for the data set.
- On the DITTO command, specify FILEIN= or FILEOUT= followed by the name of the DLBL statement. If you used the default DLBL name (SDSKIN for SAM input or SDSKOUT for SAM output), you do not need a FILEIN or FILEOUT statement.
- On the DITTO command, you can optionally specify INPUT= or OUTPUT= followed by one of the following:
 - The physical device address
 - The letter V followed by the volume serial number (*Vvolser*)
 - The programmer logical unit (*SYSnnn*)

Do not specify INPUT or OUTPUT if the SAM data set is VSAM-managed, or if you completely identified the device on the DLBL/EXTENT statement.

MVS: Specifying Sequential Input and Output without JCL

Under MVS, you can define sequential input and output without JCL. Provide the following information:

| | |
|--------------|--|
| DSNIN/OUT | The name of a sequential data set. For a partitioned data set, you also specify the member name. For example: DSNIN=dsname(member) |
| VOLSERIN/OUT | The volume serial number of the tape or disk that contains the data set, if the data set is not cataloged. |
| UNITIN/OUT | The unit that contains the data set, if the data set is not cataloged or is located on a tape. You can specify any of the following: <ul style="list-style-type: none"> • A 3-hexadecimal-digit device number • A 4-hexadecimal-digit device number prefixed by a slash (/) • A device type (generic name) • An installation-defined group name (symbolic name). |

If the data set will be used for output, you can also provide the following information:

| | |
|-------|---|
| DISP | The disposition: OLD, MOD, NEW, or CAT (for NEW,CATLG). The default is NEW. |
| SPACE | In any mode except full-screen mode, the number of primary and (optionally) secondary tracks allocated for a new data set. For example: SPACE=10 SPACE=(10,5) |

Note: You cannot use DITTO in batch mode to dynamically create an output PDS. This is possible only in full-screen mode.

In full-screen mode, you are also prompted for the following information if you specify an output data set that does not already exist:

- SMS data class
- SMS storage class
- SMS management class
- Space unit: BLKS (blocks), TRKS (tracks), CYLS (cylinders), KB, or MB
- Number of primary units
- Number of secondary units
- Average block size, if the space unit is BLKS
- Number of blocks for the directory. The panel contains this field only if you specified a member name.
- Library type: LIBRARY, PDS, or blank for the system default. The panel contains this field only if you specified a member name.
- Expiration date, in the form *yyyy.ddd*

Note: If you want to use any allocation parameters that are not supported within DITTO, you must use the TSO ALLOCATE command to allocate the data set outside DITTO. (For example, within DITTO you cannot allocate to SYSIN or SYSOUT.)

VSE: Specifying Sequential Input and Output without JCL

Under VSE, you can define sequential input and output without JCL.

Specify the following information for a VSAM-managed SAM data set:

| | |
|------------|--|
| DSNIN/OUT | The name of a sequential data set. |
| VSAMIN/OUT | Specify YES. |
| UCATIN/OUT | The user catalog, if any. |
| DISP | The disposition for an output data set: OLD or NEW. The default is NEW. |
| SPACE | The number of primary and (optionally) secondary space units allocated for a new data set. The unit is the average record length. For example: SPACE=10 SPACE=(10,5) |
| AVGSIZE | The average record length for a new data set. |

Specify the following information for a BAM-managed SAM data set:

| | |
|--------------|--|
| DSNIN/OUT | The name of a sequential data set. |
| VSAMIN/OUT | Specify NO (the default). |
| INPUT/OUTPUT | Specify one of the following: <ul style="list-style-type: none">• The physical device address• The letter V followed by the volume serial number (<i>Vvolser</i>) In full-screen mode, you can specify only a <i>volser</i> . |
| BEGIN | The begin position for an output data set. |
| END | The end position for an output data set. |

CMS: Specifying Sequential Input

Under CMS, you can use sequential input, but not sequential output. You do not use JCL to define sequential input.

Provide the following information for a sequential input file:

| | |
|-------|--|
| DSNIN | The name of a sequential data set. |
| INPUT | Specify one of the following: <ul style="list-style-type: none">• The physical device address• The letter V followed by the volume serial number (<i>Vvolser</i>) |

Record Format and Block Size

MVS Input record format is taken from the DCB. For output, you can specify the record format and block size.

VSE You can specify the record format and block size for input and output. By default, DITTO assumes an undefined record format.

CMS You can specify the record format and block size for input. By default, DITTO assumes an undefined record format.

Note: For the **SC** function (Copy sequential data to cards), you cannot specify the record format and block size of the SAM input file.

Specifying Records to be Processed

By default, DITTO functions process the entire input data set. To process a partial data set, you can:

- Specify the number of records to skip from the beginning of the data set
- Specify the number of records to process.

Printing Sequential Data

Use the **SP** or **QP** function to print a sequential data set in character or hexadecimal format. The SET parameter DUMP=UPDOWN causes hexadecimal print requests to print in hexadecimal updown format; DUMP=ACROSS causes dumps to print in hexadecimal across format. See “Formats for Displaying or Printing Data” on page 4 for an example of each format.

MVS: Printing Physical Records

By default, logical records are printed. To print physical records (under MVS):

- In full-screen mode, enter YES in the Full blocks? field.
- In line mode, respond YES to the Print physical blocks? prompt.
- In batch mode or command mode with the keyword syntax, specify MODE=U

VSE: Printing SYSLST Disk Files

Use the **SFA** function to print SYSLST type A disk files (with CCW command codes), or use the **SFD** function to print SYSLST type D disk files (with ASA control characters). The corresponding functions for SYSLST tape files are TFA and TFD.

SYSLST type A files contain CCW command codes to control how each line is printed. The SFA function assumes that any line with a length of 121 characters was produced by job control, and therefore contains ASA control characters.

SYSLST type D files contain ASA control characters in the first character of each line. The SFD function translates ASA control characters into their corresponding CCW command codes, according to the table provided in *VSE/ESA System Macro User's Guide*. For any undefined character, a space and single line are printed.

Controlling Print Output

Several SET parameters affect the printed output:

| | |
|-----------------|--|
| DATAHDR | Whether the output contains a data header for each record |
| DBCSPRT | How double-byte characters are printed |
| DUMP | Which format (updown or across) is used for hexadecimal output |
| HEADERPG | Whether the output contains a header page at the beginning |
| PAGESIZE | How many lines per page the output has |
| PAGESKIP | Whether output from each function starts on a new page |
| PRINTLEN | How many columns wide the output is |
| PRINTOUT | Where the output is sent |
| PRRTRANS | Whether non-printable characters are translated to blanks |
| RECLIMIT | Which part of each record to print |

For more information about SET parameter values, see “SET (Set Processing Options)” on page 242.

Copying Sequential Data Records

The following functions let you copy records from sequential data sets:

SS, QQ MVS or VSE: copies records from one sequential data set to another.

SV, QV Copies records from a sequential data set to a VSAM data set.

SF CMS: copies records from a sequential data set to a CMS file.

SL VSE: copies records from a sequential data set to a library member.

SC, QC Copies records from a sequential data set to cards, and optionally prints the interpreted contents on the cards. The input records must be card-image (80 characters, 81 characters under VSE, or a multiple of 80). If interpreted output is required, the card punch must have the card print feature installed.

ST, QT Copies records from a sequential data set to tape.

If you used the TS function to copy a multfile tape to a single sequential data set, you can use the ST function to copy the data set back to tape. (For more information, see “Copying Tape Data across a Network” on page 81.)

SO, QO MVS: copies sequential data to an OAM object. You can use this function to restore one or more OAM objects that were backed up in a sequential data set, as described in “Restoring OAM Objects from a Sequential or VSAM Data Set” on page 139.

MVS or VSE: Creating Records

Use the **BS** or **BQ** function to write simple test records to a sequential data set. You specify the data set that will contain the records, the number of records, and the record length. You can also specify a fill character or fill pattern, a sequence field, a record format, and a block size.

MVS: Depending on the disposition of the data set, this function creates a new data set, overwrites records in an existing data set, or adds records to the end of an existing data set.

Batch Examples Using DITTO Sequential Data Functions

Figure 63 shows a DITTO batch job that creates a new sequential data set, and then copies this data set to another sequential data set.

A DD statement for a new data set, to be created using DITTO. The example data set is a new one, to be created on DASD.

B DITTO control statement to write records to a data set.

The BS function is used to write 99 records to the data set allocated to DITIN, which is defined as new. These records are 150 characters long, are filled with random characters, and have a sequence field.

C DD statement for an output data set.

The DCB parameters for record format and block size are supplied. These are not required as the example data set already exists. The example output data set has a disposition of OLD. When this data set is used as the output data set in a DITTO copy function, as shown in **D**, the new records will be written over the old ones.

D DITTO control statement to copy a sequential data set.

The SS function is used to copy records 31-80 from the newly created PESEN.QSAM.SAMPLE data set to the PESEN.SSOUT.SAMPLE data set.

```

//S1      EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
A //DITIN  DD DSN=KEISTEW.QSAM.SAMPLE,DISP=(,CATLG),UNIT=SYSALLDA,
//        SPACE=(TRK,(1,1),RLSE),RECFM=FB,BLKSIZE=1500
C //DITOUT DD DSN=KEISTEW.SSOUT.SAMPLE,DISP=OLD,
//        RECFM=FB,BLKSIZE=3000
//SYSIN   DD *
B $$DITTO BS  OUTPUT=DITIN,RECSIZE=150,NLRECS=99,FILLCHAR=RAND,
$$DITTO   KEYLOC=145,KEYLEN=6,INCR=9999
D $$DITTO SS  INPUT=DITIN,OUTPUT=DITOUT,POSITION=30,NLRECS=50
$$DITTO  EOJ
/*
//

```

Figure 63. Batch Example Using DITTO Sequential Data Functions (MVS)

Figure 64 shows a DITTO batch job that copies a SAM file to a VSAM KSDS.

- A** IDCAMS definition of a VSAM KSDS, ending at **B**.
- C** DLBL and EXTENT statements for the SAM input file.
- D** DLBL statement for the VSAM output file.
- E** DITTO control statement to copy the SAM file to the VSAM file. The SAM input file has fixed-length records, which are to be deblocked to 80-byte records for the VSAM output. Because the NLRECS parameter is not specified, the entire SAM file will be copied.

```

// JOB SV - COPY SAM DISK FILE TO VSAM KSDS
  ASSGN SYS010,130
A // EXEC IDCAMS,SIZE=AUTO
  DELETE VSAM.KSDS CLUSTER
  DEFINE CLUSTER -
    ( NAME ( VSAM.KSDS ) -
      TRACKS ( 10, 5 ) -
      VOL ( DOSRES ) ) -
  DATA -
    ( NAME ( VSAM.KSDS.DATA ) -
      KEYS ( 4 0 ) -
      RECORDSIZE ( 30 3000 ) -
      VOL ( DOSRES ) ) -
  INDEX -
    ( NAME ( VSAM.KSDS.INDEX ) -
      IMBED )
B /*
// UPSI 1
C // DLBL CSQOUT,'CSQ.OUT',0
// EXTENT SYS010
D // DLBL KSDS,'VSAM.KSDS',,VSAM
// EXEC DITTO
E $$DITTO SV FILEIN=CSQOUT,FILEOUT=KSDS,RECFMIN=F,RECSIZE=80
  $$DITTO EOJ
/&

```

Figure 64. Batch Example Using DITTO Sequential Data Functions (VSE)

Chapter 9. Processing VSAM Data Sets

You can use the following to process VSAM data sets:

Figure 65. DITTO Functions for Processing VSAM Data Sets

| Purpose | Functions | |
|--|---------------|---------------------------|
| Get information about VSAM data sets | VDL | VSAM Data Set List |
| Change from one VSAM catalog to another | VMC | VSAM Master Catalog Setup |
| Browse a VSAM data set at your terminal | VB | VSAM Browse |
| | DVB | Data in Virtual Browse |
| Update a VSAM data set | VE | VSAM Edit |
| | VRU | VSAM Record Update |
| | VU | VSAM Update |
| Print records from VSAM data sets, in character or hexadecimal format | VP | VSAM Print |
| Copy VSAM records to cards, tape, CMS file, library member, OAM object collection, sequential data set, or VSAM data set | VC | VSAM to Card |
| | VT | VSAM to Tape |
| | VF | VSAM to CMS File |
| | VL | VSAM to Library Member |
| | VO | VSAM to Object |
| | VS, VQ | VSAM to Sequential Data |
| | VV | VSAM to VSAM |
| Copy VSAM records to or from a REXX stem variable | VX | VSAM to REXX Variable |
| | XV | REXX Variable to VSAM |
| Write records to a VSAM data set | BV | Create VSAM Data Set |

Specifying VSAM Input and Output

Functions that use VSAM input or produce VSAM output are available in all modes.

MVS or VSE: you can define VSAM input and output with or without JCL.

CMS: you can define VSAM input and output, but not with JCL. CMS/VSAM is required for all functions that use VSAM input or produce VSAM output. Make sure that all the disks you need are linked and accessed.

Note: VSAM input or output must be completely defined with JCL or completely defined through DITTO. You cannot mix JCL statements and DITTO keywords when defining input or output.

For information about working with VSAM catalogs, see Chapter 5, "Processing Catalog Entries" on page 51.

Specifying VSAM Input and Output with JCL

Under MVS or VSE, you can use job control language (JCL) to define VSAM input and output.

MVS: specify INPUT= or OUTPUT= followed by a DD or TSO ALLOC statement name (ddname). The default ddnames are VSAMIN for input, and VSAMOUT for output. Within the batch job, provide a DD statement for each ddname.

VSE: specify FILEIN= or FILEOUT= followed by the name of a DLBL statement. The default DLBL names are VDSKIN for input and VDSKOUT for output. You can also specify the password for a password-protected data set (PWDIN or PWDOUT keyword).

For VSAM output, you can also specify REUSE=YES if you want to replace any existing records in the data set (if the data set is defined with the REUSE parameter in the catalog). The default is to add new records without deleting the old records.

Specifying VSAM Input and Output without JCL

In any environment (MVS, VSE, or CMS), you can define VSAM input and output without JCL.

MVS: specify a data set name (DSNIN or DSNOUT keyword). The VSAM data set is automatically allocated. If the VSAM data set is not already defined in a catalog, use the VDL or SCS function within DITTO (or IDCAMS outside DITTO) to define the data set.

VSE: specify a data set name (DSNIN or DSNOUT keyword). You can also specify a user catalog (UCATIN or UCATOUT keyword). In command mode with the keyword syntax, you can also specify a password (PWDIN or PWDOUT keyword).

CMS: specify a data set name (DSNIN or DSNOUT keyword). You can also specify a user catalog (UCATIN or UCATOUT keyword). In command mode with the keyword syntax, you can also specify a password (PWDIN or PWDOUT keyword).

For VSAM output, you can also specify REUSE=YES if you want to replace any existing records in the data set (if the data set is defined with the REUSE parameter in the catalog). The default is to add new records without deleting the old records.

CMS: Specifying the VSAM Master Catalog

The first time you select a DITTO VSAM function under CMS, you are prompted for the physical device address and name of the VSAM master catalog that you want to work with. These values are used for the current session and all future sessions by the same user ID.

To change the master catalog that is used, use the **VMC** command. You are prompted for the physical device address and name. These values permanently replace the previously specified values.

Specifying Records to be Processed

The DITTO functions that copy VSAM records always copy logical records. Most other functions that use VSAM input let you work with either logical records or control intervals. For example, you can use the **VB** (VSAM Browse) function with control interval access to browse both data and control information.

By default, VSAM functions process the entire VSAM data set. To process a partial data set, you can:

- Specify the number of records (or control intervals) to skip from the beginning of the data set
- Specify the first key value to process, for a key-sequenced data set
- Specify the first slot value to process, for a relative record data set
- Specify the number of records (or control intervals) to process.

The maximum length of a key is 30 characters. Enclose the key in single quotation marks if it contains blanks, commas, or lowercase letters. For hexadecimal data, enter an X followed by the string enclosed in single quotation marks, for example, X'C1C2C3'.

In a VSAM data set without keys, records are processed sequentially by *address*. In a keyed data set, records are generally processed sequentially by *key*, but the browse function switches between key order and address order as required. A message is issued whenever DITTO switches from one order to the other.

Control Interval Access

In the **VB**, **VU**, **VP**, or **VRU** function, specify control interval access (CIACCESS=YES) if you want to process full control intervals, including control information.

If you specify CIACCESS=YES:

- The number you specify for *position* is the number of control intervals to skip from the beginning of the data set. If you do not specify a value, processing begins with the first control interval in the data set.
- The value you specify for *number of records* (in a print function) is interpreted as the number of control intervals to be printed. If you do not specify a value, all remaining control intervals are printed.
- When you browse control intervals of a key range data set, you may experience read errors caused by allocated but unformatted storage. To recover, enter a valid RBA.

For a path, CIACCESS=YES is ignored. For linear data sets under MVS, CIACCESS is turned on internally.

Browsing and Updating VSAM Data Sets

In full-screen mode, you can use the **VB** function to browse VSAM data sets. Under MVS, you can also use the **DVB** function to browse data-in-virtual data sets. For more information about these functions, see Chapter 3, “Browsing Data in Full-Screen Mode” on page 31.

Updating VSAM Data Sets in Full-Screen Mode

If you want to edit a VSAM data set in full-screen mode, you can use the **VE** function, as described in Chapter 4, “Editing Data in Full-Screen Mode” on page 39. You can also use the **VU** function (although VE is usually more convenient).

One case where you would want to use VU instead of VE is if you want to edit a VSAM data set using control interval access. This is supported by VU but not by VE. Also, the VU function requires less memory than VE.

If you decide to use the VU function, perform the following steps:

1. Select the VSAM update function in one of these ways:
 - Select the function from the VSAM Functions menu.
 - Enter VU on the command line of any DITTO panel.
 - Select the Update function from the Process pull-down menu in the SCS or VDL list.
 - Type update over the *beginning* of the data set name in the SCS or VDL list.
2. On the panel that appears, specify the data set name and whether you want to use control interval access. Control interval access lets you edit full control intervals, including control information; if you make incorrect changes to control information, the data set may become unusable.

The VSAM Update panel appears. On this panel, you can use all the commands and function keys that you can use in browse functions (as described in “Browsing Data” on page 32), plus additional commands for updating records.

3. Select a record that you want to update. Either move the cursor to the record, or use any of the locate commands that are available in browse functions.
4. Press the Zoom function key. This gives you a detailed view of the selected record.
5. Update the record in any of these ways:
 - Overtyping the existing text with new text.
 - Entering a column number in the Col field, and using the GET or COPY command to overwrite data in the record, starting at the specified column.

For the GET command, the data that you previously copied to the buffer with the PUT command is written. For the COPY command, you specify the columns of the data that you want to copy.

For example, if the value in the Col field is 5 and you enter COPY 10-12, the data in columns 10 to 12 replaces the data that was previously in columns 5 to 7. If you enter COPY 10, the character in column 10 replaces the character that was previously in column 5.

6. Use the Exit function key to save your changes or the Cancel function key to discard the changes. A log of your changes is printed. You can select another record to update or leave the function.

Updating One Record at a Time

In line mode, you can use the **VRU** function to change the contents of a specific VSAM record. (In full-screen mode, you can also use VRU, but you will probably find VE more convenient.)

Enter VRU, the name of the VSAM data set that contains the record, and the location of the record within the data set. You can specify the location as a record number, a key value, or a control interval number.

DITTO displays the record. If this is not the record that you want to change, exit the function and invoke it again.

DITTO prompts you for information about the character string you want to change.

After each change, you are asked whether the changes are complete. If you reply Y, the record is written to disk. A log of your changes is printed.

Printing VSAM Data Sets

Use the **VP** function to print a VSAM data set in character or hexadecimal format. The SET parameter DUMP=UPDOWN causes hexadecimal print requests to print in hexadecimal updown format; DUMP=ACROSS causes them to print in hexadecimal across format. See “Formats for Displaying or Printing Data” on page 4 for an example of each format.

Controlling Print Output

Several SET parameters affect the printed output:

| | |
|-----------------|--|
| DATAHDR | Whether the output contains a data header for each record |
| DBCSPRT | How double-byte characters are printed |
| DUMP | Which format (updown or across) is used |
| HEADERPG | Whether the output contains a header page at the beginning |
| PAGESIZE | How many lines per page the output has |
| PAGESKIP | Whether output from each function starts on a new page |
| PRINTLEN | How many columns wide the output is |
| PRINTOUT | Where the output is sent |
| PRTRANS | Whether non-printable characters are translated to blanks |
| RECLIMIT | Which part of each record to print |

For more information about SET parameter values, see “SET (Set Processing Options)” on page 242.

Copying VSAM Records

The following functions let you copy VSAM records:

- VV** Copies records from a VSAM data set to another VSAM data set.
- VS, VQ** MVS or VSE: copies records from a VSAM data set to a sequential data set.
- VF** CMS: copies records from a VSAM data set to a CMS file.
- VC** Copies records from a VSAM data set to cards, and optionally prints the interpreted contents on the cards. The input records must be card-image (80 characters, 81 characters under VSE, or a multiple of 80). The card punch must have the card print feature installed for interpreted output.
- VL** VSE: copies records from a VSAM data set to a library.
- VO** MVS: copies VSAM records to an OAM object. You can use this function to restore one or more OAM objects that were backed up in a VSAM data set, as described in “Restoring OAM Objects from a Sequential or VSAM Data Set” on page 139.
- VT** Copies records from a VSAM data set to tape.

If you used the TV function to copy a multifile tape to a single VSAM entry-sequenced data set, you can use the VT function to copy the data set back to tape. (For more information, see “Copying Tape Data across a Network” on page 81.)
- VX** Copies VSAM records to a REXX stem variable, if invoked from a REXX procedure. The **XV** function copies a REXX stem variable back to a VSAM data set.

Creating VSAM Records

Use the **BV** function to write fixed-length records to a VSAM data set. You specify the data set that will contain the records, the number of records, and the record length. You can also specify a fill character or fill pattern and a sequence field.

Examples of DITTO VSAM Functions

Figure 66 on page 115, Figure 67 on page 115, and Figure 68 on page 115 show an MVS batch job, a VSE batch job, and a CMS REXX procedure that work with VSAM data sets.

- A** MVS: DD statement for a VSAM input data set.

VSE: DLBL statement for a VSAM input data set.
- B** DITTO control statement to print, in character format, records 13 and 14 from a VSAM ESDS (a data set without keys). The statement refers to an MVS DD statement or VSE DLBL statement.
- C** DITTO control statement to print, in hexadecimal format, 10 records from a VSAM KSDS (a data set with keys). The record with key 025000, or the next greater key, is the first record printed. The statement refers to the data set directly; no JCL is needed.


```

//S1      EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
A //TSTESDS DD DSN=DITTO.ESDS.DSET,DISP=SHR
//SYSIN   DD *
B $$DITTO VP INPUT=TSTESDS,NLRECS=2,FORMAT=CHAR,POSITION=12
C $$DITTO VP DSNIN=DITTO.SAMP.KSDS,NLRECS=10,FORMAT=HEX,KEY=025000
$$DITTO EOJ
/*
//

```

Figure 66. Batch Example Using DITTO VSAM Functions (MVS)

```

// JOB VDVP - PRINT RECORDS FROM 2 VSAM FILES
// UPSI 1
A // DLBL TSTESDS,'DITTO.ESDS.DSET',,VSAM
// EXEC DITTO
B $$DITTO VP FILEIN=TSTESDS,NLRECS=2,FORMAT=CHAR,POSITION=12
C $$DITTO VP DSNIN=DITTO.SAMP.KSDS,NLRECS=10,FORMAT=HEX,KEY=025000
$$DITTO EOJ
/&

```

Figure 67. Batch Example Using DITTO VSAM Functions (VSE)

```

/* REXX - Print records from a VSAM data set      */
/* Define the VSAM Master Catalog                 */
"DITTO $VMC CATALOG=VSAM.MASTER.CATALOG,UNIT=144"

/* Link and access all data and ucat disks        */
/* (DITTO links the mcat disk dynamically)        */
address CMS "LINK VSESP2 14C 14C RR"
address CMS "LINK VSESP2 14D 14D RR"
address CMS "ACCESS 14C M"
address CMS "ACCESS 14D N"

/* Print ten records                              */
C "DITTO $VP DSNIN=PRIVATE.KSDS,",
    "UCATIN=VSESP.USER.CATALOG,",
    "KEY=025000,NLRECS=10"
exit

```

Figure 68. Command Mode Example Using DITTO VSAM Functions (CMS)

Chapter 10. CMS: Processing CMS Files

You can use the following functions to process CMS files:

Figure 69. DITTO Functions for Processing CMS Files

| Purpose | Functions | |
|---|-------------|-----------------------------|
| Browse a CMS file | FB | CMS File Browse |
| Print a CMS file in character or hexadecimal format | FP | CMS File Print |
| Copy from a CMS file to tape or VSAM file | FT | CMS File to Tape |
| | FV | CMS File to VSAM |
| Copy from a CMS file to a library member or sequential file (CMS/VSE mode only) | FL | CMS File to Library Member |
| | FS | CMS File to Sequential Data |
| Dump CMS files to tape | DUMP | Dump CMS Files to Tape |
| Load from tape to a CMS file | LOAD | Load Files from Tape |

Specifying CMS Input and Output

If you are running DITTO under CMS, you can use CMS files as input or output. You identify each file as follows: *filename filetype filemode*, where *filemode* is optional. For output, the default filemode is A; for input, the default is to use the normal minidisk search order. The entire file ID must be enclosed in single quotation marks.

For a file output function, you can specify a record format of F (fixed) or accept the default (variable). You can also specify that output records are appended to the end of an existing CMS file. By default, any existing file with the specified file ID is replaced.

You can use shared files in the same way as other CMS files, by means of an accessed filemode. You cannot specify directories or define access to shared files within DITTO.

Browsing CMS Files

In full-screen mode, you can use the **FB** function to browse a CMS file. This function reads only the data for the current screen into storage, so it is especially useful if you are working with dump files or other large files. You can use this function with either ASCII or EBCDIC data.

For more information about this function, see Chapter 3, "Browsing Data in Full-Screen Mode" on page 31.

Printing CMS Files

Use the **FP** function to print a CMS file in character or hexadecimal format. The SET parameter DUMP=UPDOWN causes hexadecimal print requests to print in hexadecimal updown format; DUMP=ACROSS causes them to print in hexadecimal across format. See "Formats for Displaying or Printing Data" on page 4 for an example of each format.

You can specify the number of records to skip from the beginning of the file, and the number of records to print. You can also specify a record length, if you want to deblock the physical records and print the logical records separately. The record length of the CMS file does not need to be a multiple of the record length that you specify; any remainder is printed as a record.

Controlling Print Output

Several SET parameters affect the printed output:

| | |
|-----------------|--|
| DATAHDR | Whether the output contains a data header for each record |
| DBCSPRT | How double-byte characters are printed |
| DUMP | Which format (updown or across) is used for hexadecimal output |
| HEADERPG | Whether the output contains a header page at the beginning |
| PAGESKIP | Whether output from each function starts on a new page |
| PRINTLEN | How many columns wide the output is |
| PRINTOUT | Where the output is sent |
| PRTRANS | Whether non-printable characters are translated to blanks |
| RECLIMIT | Which part of each record to print |

For more information about SET parameter values, see “SET (Set Processing Options)” on page 242.

Copying CMS Files

The following functions let you copy CMS files:

FV Copies records from a CMS file to a VSAM file.

FT Copies records from a CMS file to tape.

If you used the TF function to copy a multiframe tape to a single CMS file, you can use the FT function to copy the file back to tape. (For more information, see “Copying Tape Data across a Network” on page 81.)

You can specify the number of records to skip from the beginning of the file, and the number of records to copy.

Copying Files in CMS/VSE Mode

If you invoke a full-screen VSE/DITTO session under CMS (as described in “CMS/VSE Mode: Starting VSE/DITTO from CMS” on page 14), you can use the following additional copy commands:

FL Copies records from a CMS file to a library member.

FS Copies records from a CMS file to a sequential file.

You can specify the number of records to skip from the beginning of the file, and the number of records to copy.

Dumping and Restoring CMS Files

You can use the **DUMP** function to store several CMS files on tape. You can use the **LOAD** function to restore files from any of the following:

- A dump tape that was created with the DITTO DUMP command
- A minidisk from a DASD Dump Restore (DDR) tape or file
- A minidisk from a fullpack DDR tape or file (or from a multivolume backup tape generated by one DDR run with continuation tapes, or several consecutive DDR runs)

Dumping Files to Tape

The DUMP function dumps the files that you specify to the tape that you specify, then writes a tape mark. If the end of the tape is reached, DITTO prompts you to mount a new tape.

The files that you dump are listed on the terminal or in the batch output. In command mode, a return code is also issued. Return code 0 indicates successful completion.

For example, the following control statements dump all the files on your A disk to a tape at logical address 181:

```
DITTO $REW OUTPUT=181
DITTO $DUMP INFIL='* * A',OUTPUT=181
```

Loading Files from a DITTO Dump Tape

To load files from a DITTO dump tape, you specify the files that you want to load and the logical address of the tape.

You can limit the files that are loaded in any of the following ways:

- You must specify a file ID for the file or files that you want to load, in the format *filename filetype filemode*, where *filemode* is optional, and the entire file ID is enclosed in single quotation marks. Within the file ID, you can include a percent sign (%) to represent exactly one character, or an asterisk (*) to represent any number of characters (or no characters).
- You can specify the number of files that you want to load. The default is 1.
- You can specify SCAN, and a file ID enclosed in single quotation marks. DITTO skips all preceding files and starts loading files at the specified file.
- You can specify SKIP, and a file ID enclosed in single quotation marks. DITTO skips the file you specify and all preceding files, and starts loading files *after* the specified file.

To list all the files without loading any files, specify either SCAN or SKIP with the file ID '* *'.

Loading Files from a DDR or Fullpack DDR Tape or File

When you load from a DDR or fullpack DDR tape or file, you specify the minidisk (on the backup tape) that you want to load. This minidisk is loaded into a staging file. The files that you specify are then copied from the staging file to disk. You can optionally keep this staging file and use it for subsequent load operations. The staging file takes up approximately 5% more storage than the original minidisk.

You can load files from a minidisk that originally existed on a different type of disk. For example, you can dump an FBA disk to tape with DDR and later use the DITTO LOAD command to restore a minidisk from the DDR tape to a CKD disk.

To load files from a DDR tape, you specify the files that you want to load, DDR or FULLDDR (whichever applies), the volume serial number of the minidisk, the address of the minidisk (for fullpack DDR only), and the location of the DDR input (a tape address or a CMS file ID). You can also specify that you want to keep the staging file.

You can limit the files that are loaded in any of the following ways:

- You must specify a file ID for the file or files that you want to load, in the format *filename filetype filemode*, where *filemode* is optional, and the entire file ID is enclosed in single quotation marks. Within the file ID, you can include a percent sign (%) to represent exactly one character, or an asterisk (*) to represent any number of characters (or no characters).
- You can specify SCAN, and a file ID enclosed in single quotation marks. DITTO skips all preceding files and starts loading files at the specified file.
- You can specify SKIP, and a file ID enclosed in single quotation marks. DITTO skips the file you specify and all preceding files, and starts loading files *after* the specified file.

To list all the files without loading any files, specify SCAN or SKIP, followed by two asterisks (* *) enclosed in single quotation marks.

For example, if you want to load several files from a DDR tape you could enter the following commands in CMS:

1. Rewind the tape at logical address 181:

```
DITTO $REW INPUT=181
```

2. List all the files on the USR191 minidisk from the DDR tape at address 181, and keep the staging file:

```
DITTO $LOAD OUTFILE='* * A' TYPE=DDR, VOLSER=USR191, INPUT=181,  
KEEP=YES, SCAN='* *'
```

No files are actually loaded.

3. Load the file named PROFILE XEDIT from the staging file for minidisk USR191 to your A disk:

```
DITTO $LOAD OUTFILE='PROFILE XEDIT A', TYPE=DDR, VOLSER=USR191
```

Because a staging file for USR191 already exists on disk, you do not need to specify a tape address.

4. Load all the files with file type SCRIPT from the staging file for minidisk USR191 to your A disk:

```
DITTO $LOAD OUTFILE='* SCRIPT A', TYPE=DDR, VOLSER=USR191
```


Chapter 11. VSE: Processing Library Members

You can use the following functions to process VSE library members:

Figure 70. DITTO Functions for Processing Library Members

| Purpose | Functions | |
|---|-------------|-----------------------------------|
| List members in a library | LDL | Library Directory List |
| List status information for a library | LSL | Library Status List |
| Browse a library member at your terminal | LB | Library Member Browse |
| Update a library member interactively | LE | Library Member Edit |
| | LU | Library Member Update |
| Print a library member, in character or hexadecimal format | LP | Library Member Print |
| Copy a library member to cards, tape, sequential data, or a VSAM data set | LC | Library Member to Card |
| | LT | Library Member to Tape |
| | LF | Library Member to CMS File |
| | LS | Library Member to Sequential Data |
| | LV | Library Member to VSAM |
| Copy library data to or from a REXX stem variable | LX | Library Member to REXX Variable |
| | XL | REXX Variable to Library Member |
| Copy library members to a different library, or to different names within a library | LL | Library Member to Library Member |
| Rename library members | LR | Library Member Rename |
| Delete library members | LDEL | Library Member Delete |

Specifying Library Input and Output

Under VSE, you can use a member in a VSE library as input or output. To do this, you identify the library and member as follows:

1. Specify the library name, followed by a period and the sublibrary name (*library.sublibrary*).

DITTO looks in the standard label area for the library name that you specify. If the library name is not found (in line mode or full-screen mode), you are prompted for:
 - The data set name of the library
 - The volume serial number (for a library in BAM-managed space)
 - An optional user catalog name (for a library in VSAM-managed space)
2. Specify the member name, followed by a period and the type (*name.type*).
3. For a library output function, you can specify a record format of F (fixed, the default) or S (string). String format is always used if the member type is PHASE.

Listing the Members in a Library

You can use the **LDL** (Library Directory List) function to display or print a list of library members. The list can be based on library name, member name, and creation date.

The information in the list includes:

- Member name
- Member type
- Library name
- Sublibrary name
- Member size
- Used library blocks
- Last update date and time
- Creation date and time
- Whether the member is controlled by MSHP
- Whether the member contains SYSIPT data
- The LOCKID

If a library member has never been updated, the Last Update column shows the creation date and time followed by an asterisk (*).

You can sort the list by member name, type, last update date, and size.

In line mode, command mode, or batch mode, you can use LDL to print a list of library members. In full-screen mode, you can also use LDL to display a full-screen library member list; from this list, you can select the member that you want to work with.

If no library is specified, all libraries stored in the system standard labels are scanned.

Using the LDL Function in Full-Screen Mode

1. Select the LDL option from the Librarian Functions menu, or enter LDL on the command line of any panel in DITTO.
2. On the panel that appears, you can specify some or all of the following to limit the members that are listed:
 - Up to 8 library names or generic library names. If you do not specify any library names, all of the libraries stored in the system standard labels are scanned.
 - A member name or generic member name
 - The earliest update date
 - The latest update date

You can also specify a sort order, and whether you want the list to be displayed or printed.
3. DITTO lists the selected members.

```

Process  View  Options  Help
-----
DITTO/ESA for VSE      LDL - Library Directory List                Line 21 of 61

Library PRD2

--Library members sorted by NAME  ---Bytes--- Library  ---Last Update---
-Member- --Type-- -LIBID- -Sublib- --Records--- -blocks ---Date--- --Time--
ATCSTR00 B      PRD2  CONFIG      19 R      1 1993-03-26 16:44.17*
ATCSTR00 SAVE  PRD2  CONFIG      19 R      1 1993-03-26 16:44.16*
CPUVAR1  PROC   PRD2  SAVE       62 R      2 1993-03-26 10:13.32*
DFHRDTSP PHASE  PRD2  CONFIG     396 B      1 1993-03-26 16:43.35
DFHCTSP  PHASE  PRD2  CONFIG    5528 B      6 1993-03-26 16:43.35
DTRCICST PROC   PRD2  SAVE        9 R      1 1993-03-26 10:13.33*
DTRICCF  PROC   PRD2  SAVE        1 R      1 1993-03-26 10:13.33*
DTRINFOA PROC   PRD2  SAVE        2 R      1 1993-03-26 10:13.33*
DTRPOWR  PROC   PRD2  SAVE        3 R      1 1993-03-26 10:13.34*
DTRTERM1 PHASE  PRD2  CONFIG     140 B      1 1993-03-26 16:44.04
DTSECTAB PHASE  PRD2  SAVE     3520 B      4 1993-07-05 15:33.32
DTSECTRC A      PRD2  SAVE       174 R      9 1993-06-08 14:24.45

Command ==> _____ Scroll CSR
F1=Help F2=Browse F3=Exit F4=Process F5=Rgtleft F6=Rfind F7=Bkwd F8=Fwd
F10=Actions F11=CRetrieve F12=Cancel

```

Figure 71. Library Directory List

By using the Rgtleft function key you can switch between two display formats to see all the information you would get on a full-width printout. The second display format is shown in Figure 72 on page 126.

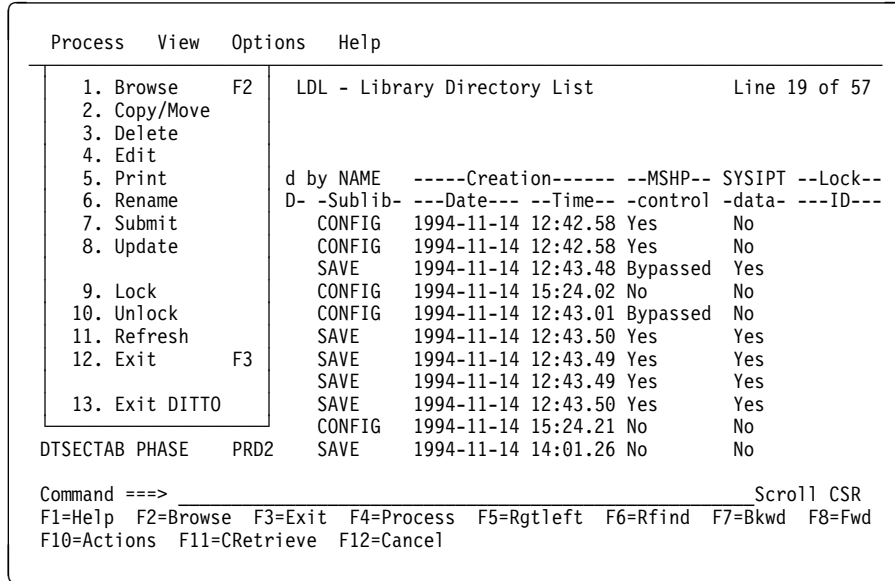


Figure 72. Library Directory List (Alternative Display)

To move forward or backward through the library member list, use the Fwd or Bkwd function key, or use the locate command. You can also enter a value in the Scroll field. (In DITTO, all full-screen list functions and full-screen browse functions use the same basic methods for locating data. These methods are described in Chapter 3, "Browsing Data in Full-Screen Mode" on page 31.)

To work with a library member, move the cursor to the member name, and press the Process function key. (If the cursor is outside the list area, you will work with the first entry in the list.) The selected line is highlighted in another color. A pull-down menu appears below the action bar. Select an option on the pull-down menu to browse, delete, edit, print, rename, or update the library member, or submit it to the POWER reader queue.

You can also type the following line commands over the beginning of the library member name:

- browse** Invokes LB
- copy** Invokes LL
- delete** Invokes LDEL
- edit** Invokes LE (for a fixed format member) or LU (for a string format member)
- lock** Locks the member
- print** Invokes LP
- rename** Invokes LR
- submit** Submits the library member to the POWER spool queue
- unlock** Unlocks the member
- update** The same as **edit**

Listing Status Information for a Library

You can use the **LSL** (Library Status List) function to display or print status information and a list of sublibraries for a library.

For the library as a whole, the following information is listed:

- Space management
- Number of extents
- Number of sublibraries
- Creation date and time
- Library blocks used, free, and delayed
- Block size

For each sublibrary, the following information is listed:

- Sublibrary name
- Number of members
- Percentage of library space occupied
- Library blocks used, delayed, and limit
- Whether space reuse is automatic
- Creation date and time

For each extent, the following information is listed:

- Volume serial
- Extent begin and end
- Device type and architecture

In line mode, command mode, or batch mode, you can use LSL to print this information. In full-screen mode, you can also use LSL to display the information on a full-screen panel.

Using the LSL Function in Full-Screen Mode

1. Select the LSL option from the Librarian Functions menu, or enter LSL on the command line of any panel in DITTO.
2. On the panel that appears, enter the name of the library. You can also specify whether you want the list to be displayed or printed.
3. DITTO displays status information and lists the sublibraries.

```

Process  View  Options  Help
-----  -
DITTO/ESA for VSE          LSL - Library Status List          Line 1 of 8

Library PRD2                DSNAME VSE.PRD2.LIBRARY
Space management. .: VSAM          Total blocks: 21855 (100%)
Number of extents .: 1            Used blocks : 121 ( 1%)
Total sublibraries : 8           Free blocks : 21733 ( 99%)
Creation date . . .: 1992-07-30   Delayed blks: 1 ( 0%)
Creation time . . .: 11:25.01     Block size .: 1024 bytes

-Sublib- -Total- % Lib ----Library Blocks----- --Space-- -----Creation-----
--name-- members space --Used- Delayed -Limit- --reuse-- ---Date--- --Time--
AFP      0      0%      1      0      NONE  AUTOMATIC 1992-07-30 11:25.09
COMM     0      0%      1      0      NONE  AUTOMATIC 1992-07-30 11:25.07
COMM2    0      0%      1      0      NONE  AUTOMATIC 1992-07-30 11:25.08
CONFIG   24     0%     47     0      NONE  AUTOMATIC 1992-07-30 11:25.02
DBASE    0      0%      1      0      NONE  AUTOMATIC 1992-07-30 11:25.05
PROD     11     0%     12     1      NONE  AUTOMATIC 1992-07-30 11:25.04
SAVE     35     0%     53     0      NONE  AUTOMATIC 1992-07-30 11:25.03
SQL320   0      0%      1      0      NONE  AUTOMATIC 1992-07-30 11:25.10
Command ==>                               Scroll CSR
F1=Help F2=Directory F3=Exit F4=Process F5=Extents F7=Bkwd F8=Fwd
F10=Actions F11=CRetrieve F12=Cancel

```

Figure 73. Library Status List

To move forward or backward through the sublibrary list, use the Fwd or Bkwd function key. You can also enter a value in the Scroll field.

To display the contents of a sublibrary, move the cursor to the sublibrary name and press the Directory function key. (You can also type dir over the beginning of the sublibrary name and press Enter.) This invokes the LDL function.

To see which disk extents are used by the library, press the Extents function key. The disk extents display appears, as shown in Figure 74.

```

Process  View  Options  Help
-----  -
1. Directory F2          LSL - Library Status List          Line 1 of 7
2. Browse
3. Edit
4. Print
5. Update
6. Extents
7. Sublibs
8. Exit      F3
9. Exit DITTO

          DSNAME VSE.PRD1.LIBRARY
          SAM          Total blocks: 96255 (100%)
          994-11-14   Used blocks : 81050 ( 84%)
          1:59.01    Free blocks : 15205 ( 16%)
          Delayed blks: 0 ( 0%)
          Block size .: 1024 bytes

--Extent----- -----Device-----
egin- ---End-- -Type- -Architecture-
26-00 271-14 3380  CKD
2  DOSRES 276-00 298-14 3380  CKD
3  DOSRES 299-00 321-14 3380  CKD
4  DOSRES 322-00 344-14 3380  CKD
5  DOSRES 345-00 367-14 3380  CKD
6  DOSRES 368-00 390-14 3380  CKD
**** End of data ****
Command ==>                               Scroll CSR
F1=Help F2=Directory F3=Exit F4=Process F5=Sublibs F7=Bkwd F8=Fwd
F10=Actions F11=CRetrieve F12=Cancel

```

Figure 74. Library Status List (Disk Extents Display)

To work with the physical disk extents that are listed, press the Process function key. A pull-down menu appears below the action bar. Select an option on the pull-down menu to browse, print, or update the disk extents.

You can also type the following line commands over the beginning of the extent number:

browse Invokes the DB function
edit Invokes the DTE function
print Invokes the DP function
update Invokes the DU function

To switch back to the sublibraries display, press the SubLibs function key.

Browsing and Updating Library Members in Full-Screen Mode

You can use the **LB** function to browse library members. For more information about this function, see Chapter 3, “Browsing Data in Full-Screen Mode” on page 31.

If you want to edit a library member in full-screen mode, you can use the **LE** or **LU** function as follows:

1. Select the library member edit or update function in one of these ways:
 - Select the function from the Librarian Functions menu.
 - Enter LE or LU on the command line of any DITTO panel.
 - Select the Edit or Update function from the Process pull-down menu in the LDL list.
 - Type edit or update over the *beginning* of the library member name in the LDL list.

If the library member is fixed-format, the Library Edit panel appears. Follow the instructions in Chapter 4, “Editing Data in Full-Screen Mode” on page 39.

If the library member is string-format, the Library Update panel appears. On this panel, you can use all the commands and function keys that you can use in browse functions (as described in “Browsing Data” on page 32), plus additional commands for updating library members.

2. Update a string-format member in any of these ways:
 - Overtyping the existing text with new text.
 - Enter a column number in the Col field, and use the GET or COPY command to overwrite data, starting at the specified column.

For the GET command, the data that you previously copied to the buffer with the PUT command is written. For the COPY command, you specify the columns that you want to copy.

For example, if the value in the Col field is 5 and you enter COPY 10-12, the data in columns 10 to 12 replaces the data that was previously in columns 5 to 7. If you enter COPY 10, the character in column 10 replaces the character that was previously in column 5.

3. Use the Exit function key to save your changes or the Cancel function key to discard the changes.

Printing Library Members

Use the **LP** function to print a library member in character or hexadecimal format. The **SET** parameter **DUMP=UPDOWN** causes hexadecimal print requests to print in hexadecimal updown format; **DUMP=ACROSS** causes them to print in hexadecimal across format. See “Formats for Displaying or Printing Data” on page 4 for an example of each format.

You can also specify the number of records or bytes to skip from the beginning of the member and the number of records or bytes to print.

Controlling Print Output

Several **SET** parameters affect the printed output:

| | |
|-----------------|--|
| DATAHDR | Whether the output contains a data header for each record |
| DBCSPRT | How double-byte characters are printed |
| DUMP | Which format (updown or across) is used for hexadecimal output |
| HEADERPG | Whether the output contains a header page at the beginning |
| PAGESKIP | Whether output from each function starts on a new page |
| PRINTLEN | How many columns wide the output is |
| PRINTOUT | Where the output is sent |
| PRTRANS | Whether non-printable characters are translated to blanks |
| RECLIMIT | Which part of each block to print |

For more information about **SET** parameter values, see “**SET** (Set Processing Options)” on page 242.

Copying Library Members

The following functions let you copy library members:

- LL** Copies members to a different library, a different sublibrary, or different names within the same library and sublibrary. You can copy one member, or specify a generic member name to copy multiple members.
- LS** Copies a library member to a sequential data set.
- LF** Copies a library member to a CMS file. This command is available only if you have invoked full-screen VSE/DITTO from CMS (as described in “CMS/VSE Mode: Starting VSE/DITTO from CMS” on page 14).
- LV** Copies a library member to a VSAM file.
- LC** Copies a library member to cards, and optionally prints the interpreted contents on the cards.
- LT** Copies a library member to a tape file.
- LX** Copies data from a library member to a REXX stem variable, if invoked from a REXX procedure. You can specify the number of records or bytes to skip from the beginning of the member and the number of records or bytes to copy. The **XL** function copies a REXX stem variable back to a library member.

DITTO copies library members in fixed format (for example, assembler source) using fixed, 80 byte record format. A library member in string format (for example, a phase) is copied as a single record, unless you specify a block size to be used

for deblocking. The record format is always undefined. When copying to a library member, specify the format of the member (fixed or string).

If you copy a library member that is a phase, DITTO also copies directory information. This information is used if you later use DITTO to copy the phase back to a library.

Deleting or Renaming Library Members

Use the **LDEL** function to delete library members. Use the **LR** function to rename library members. You can process one member, or specify a generic member name to process multiple members.

Examples of DITTO Library Functions

Figure 75 and Figure 76 show a batch job and a REXX procedure that copy a VSAM file to a library, replacing an existing library member. The library member is then printed.

- A** DLBL statement for a VSAM input file.
- B** DITTO control statement to delete an existing library member.
- C** DITTO control statement to copy a VSAM file to a library (using the name of the previously deleted library member).
- D** DITTO control statement to print the new library member.

```

// UPSI 1
A // DLBL TSTESDS, 'DITTO.ESDS.DSET', ,VSAM
// EXEC DITTO
B $$DITTO LDEL LIBIN=PRD2.SAVE, MEMBERIN=PW1XL.PROC
C $$DITTO VL FILEIN=TSTESDS, LIBOUT=PRD2.CONFIG,
    $$DITTO MEMBEROUT=PW1XL.PROC
D $$DITTO LP LIBIN=PRD2.SAVE, MEMBERIN=PW1XL.PROC
    $$DITTO EOJ
/&

```

Figure 75. Batch Example Using DITTO Library Functions

```

/* REXX */
/* Example: Execute Library Commands from a REXX procedure */
CALL 'DITSETUP' /* setup DITTOs REXX interface */
ADDRESS 'DITTO' /* pass commands to DITTO */
B DITTO '$LDEL LIBIN=PRD2.SAVE, MEMBERIN=PW1XL.PROC'
C DITTO '$VL DSNIN=DITTO.ESDS.DSET, ',
    'LIBOUT=PRD2.CONFIG, MEMBEROUT=PW1XL.PROC'
D DITTO '$LP LIBIN=PRD2.SAVE, MEMBERIN=PW1XL.PROC'

```

Figure 76. Command Mode Example Using DITTO Library Functions

Library

Chapter 12. MVS: Processing OAM Objects

You can use the following functions to work with OAM objects:

Figure 77. DITTO Functions for Processing OAM Objects

| Purpose | Functions | |
|--|----------------------------|---|
| List objects in an OAM object collection or optical disk | ODL | Object Directory List |
| Browse an OAM object at your terminal | OB | Object Browse |
| Print an OAM object, in character or hexadecimal format | OP | Object Print |
| Back up OAM objects in a sequential or VSAM data set | OS, OQ OV | Object to Sequential Data Object to VSAM |
| Restore OAM objects from a sequential or VSAM data set, or create a new object by copying a data set | SO, QO VO | Sequential Data to Object VSAM to Object |
| Copy an OAM object to a different collection, or to a different name within a collection | OO | Object to Object |
| Update an OAM object interactively | OU | Object Update |
| Erase an OAM object | OE | Object Erase |

An Introduction to the Object Access Method

Under MVS, various Object Access Method (OAM) functions are available. This section provides a brief introduction to OAM. For more information, see *DFSMS/MVS OAM Application Programmer's Reference*.

In OAM, an *object* is a stream of bytes with a name. The content, format, and structure of that byte stream are unknown to OAM. For example, an object could be a compressed scanned image, or coded data. There is no concept of individual records within an object.

A *collection* is a group of objects that typically have similar performance, availability, backup, retention, and class transition characteristics. Every object must be assigned to a collection. Object names must be unique within a collection; the same object name can be used in several different collections.

OAM is an access method of MVS/DFP™, the base for DFSMS. OAM uses the SMS-supplied hierarchy definition and management parameters to place user-accessible objects anywhere in the storage hierarchy. The hierarchy consists of DASD, library-resident optical volumes, and shelf-resident optical volumes. The location of an object in the hierarchy is unknown to the user; device-dependent information (such as track size) is not required.

For each object, the user can specify:

| | |
|-------------------------|---|
| Storage Class | The level of service for the object, independent of the physical device or medium containing the object |
| Management Class | Backup, retention, and class transition characteristics for the object |

The storage class and management class may be overridden by your installation defaults.

Listing OAM Objects

You can use the **ODL** (Object Directory List) function to display or print a list of OAM objects. The list can be based on collection name, object name, disk location, and creation date.

The information in the list includes:

- Object name
- Object size
- Object location (DASD or optical)
- Volume serial number (if known)
- Creation and expiration dates
- Storage class
- Management class

You can sort the list by object name, size, creation date, or expiration date.

In line mode or batch mode, you can use ODL to print a list of objects. In full-screen mode, you can also use ODL to display a full-screen object list; from this list, you can select the object that you want to work with. In command mode, entering ODL invokes the full-screen mode ODL function.

Using the ODL Function in Full-Screen Mode

1. Select the ODL option from the OAM Functions menu, or enter ODL on the command line of any panel in DITTO.
2. On the panel that appears, you specify some or all of the following to limit the objects that are listed:
 - A collection name
 - An object name or generic object name
 - A *limit* value: DASD (objects on DASD only), OPTICAL (objects on optical disks only), VOLUME (objects on the specified optical disk only). If you specify VOLUME, you must also specify the volser.
 - The earliest creation date
 - The latest creation date

You can also specify a sort order, and whether you want the list to be displayed or printed.

You must specify the collection name, the volser, or both. The other values are optional.

Note: To see a list of object collections, use the SCS function as described in Chapter 5, "Processing Catalog Entries" on page 51.

3. DITTO lists the selected OAM objects.

```

Process  View  Options  Help
-----
DITTO/ESA for MVS          ODL - Object Directory List          Line 1 of 55
Collection SYSU.OAM.CLLCT000_____ SC OLIBRARY MC OPT6D___

---- Object Name ---- sorted by NAME ----- ---Creation---- -Object- Location
...5...10...15...20...25...30...35...40.... ---Date--- -Time --Size-- - VOLSER
BOURDAGE.D1                1992-03-09 22:40      512 D  N/A
BOURDAGE.IMOBJ.TT01121     1992-02-06 14:28     15400 D N/A
BOURDAGE.IMOBJ.TT01122     1992-02-06 14:28     17160 D N/A
BOURDAGE.IMOBJ.TT01123     1992-02-06 14:28     2760 D  N/A
BOURDAGE.IMOBJ.TT01124     1992-02-06 14:28     11880 D N/A
BOURDAGE.IMOBJ.TT01125     1992-02-06 14:28     4080 D  N/A
BOURDAGE.IMOBJ.TT01126     1992-02-06 14:28     6720 D  N/A
BOURDAGE.IMOBJ.TT01127     1992-02-06 14:28     9360 D  N/A
BOURDAGE.IMOBJ.TT01128     1992-02-06 14:28    10560 D N/A
BOURDAGE.IMOBJ.T8X331      1992-02-06 14:28    10800 D N/A
BOURDAGE.IMOBJ.T8X332      1992-02-06 14:28     8400 D  N/A
BOURDAGE.IMOBJ.T8X333      1992-02-06 14:28     7200 D  N/A

Command ==>_____ Scroll CSR
F1=Help  F2=Browse F3=Exit F4=Process F5=Rgtleft F6=Rfind F7=Bkwd F8=Fwd
F10=Actions F11=CRetrieve F12=Cancel

```

Figure 78. Object Directory List

By using the Rgtleft function key you can switch between two display formats to see all the information you would get on a full-width printout. The second display format is shown in Figure 79 on page 136.

```

Process  View  Options  Help
-----  -
- 1. Browse  F2      ODL - Object Directory List           Line 6 of 55
  2. Print
  3. Erase   LCT000          SC OLIBRARY MC OPT6D
  4. Update
  5. CLID List
  6. Refresh
  7. Exit    F3

sorted by NAME  ----- Storage  Managmnt  Expiration  Coll
.25...30...35...40.... --Class- --Class- ---Date--- -ID-
1
BOURDAGE.IMOBJ.TT01122      OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.TT01123      OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.TT01124      OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.TT01125      OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.TT01126      OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.TT01127      OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.TT01128      OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.T8X331       OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.T8X332       OLIBRARY OPT6D  0001-01-01  1
BOURDAGE.IMOBJ.T8X333       OLIBRARY OPT6D  0001-01-01  1

Command ==>                               Scroll CSR
F1=Help F2=Browse F3=Exit F4=Process F5=Rgtleft F6=Rfind F7=Bkwd F8=Fwd
F10=Actions F11=CRetrieve F12=Cancel

```

Figure 79. Object Directory List (Alternative Display)

To move forward or backward through the object list, use the Fwd or Bkwd function key, or use the locate command. You can also enter a value in the Scroll field. (In DITTO, all full-screen list functions and full-screen browse functions use the same basic methods for locating data. These methods are described in Chapter 3, "Browsing Data in Full-Screen Mode" on page 31.)

To work with an object, move the cursor to the object name, and press the Process function key. (If the cursor is outside the list area, you will work with the first entry in the list.) The selected line is highlighted in another color. A pull-down menu appears below the action bar. Select an option on the pull-down menu to browse, print, erase, or update the object.

If the list contains objects from more than one collection (from the same volser), the Coll ID column contains a number for each collection. The first collection has the number 1, and so on. To see the collection name that corresponds to each collection number, press the Process function key and then select CLID List.

You can also type the following line commands over the beginning of the object name:

```

browse
print
erase
update

```

Browsing and Updating OAM Objects in Full-Screen Mode

You can use the **OB** function to browse OAM objects. For more information about this function, see Chapter 3, "Browsing Data in Full-Screen Mode" on page 31.

You can use the **OU** function to update OAM objects interactively.

To update OAM objects, perform the following steps:

1. Select the object update function in one of these ways:
 - Select the function from the OAM Functions menu.
 - Enter OU on the command line of any DITTO panel.
 - Select the Update function from the Process pull-down menu in the ODL list.
 - Type update over the *beginning* of the object name in the ODL list.

The Object Update panel appears. On this panel, you can use all of the commands and function keys that you can use in browse functions (as described in “Browsing Data” on page 32), plus additional commands for updating objects.

2. Update the object in any of these ways:
 - Overtyping the existing text with new text.
 - Entering a byte number in the Col field, and using the GET or COPY command to overwrite data in the object, starting at the specified position.

For the GET command, the data that you previously copied to the buffer with the PUT command is written. For the COPY command, you specify the bytes of the object that you want to copy.

For example, if the value in the Col field is 5 and you enter COPY 10-12, the data in bytes 10 to 12 replaces the data that was previously in bytes 5 to 7. If you enter COPY 10, the character in column 10 replaces the character that was previously in byte 5.

3. Use the Exit function key to save your changes or the Cancel function key to discard the changes.

Printing OAM Objects

Use the **OP** function to print an OAM object in character or hexadecimal format. The SET parameter DUMP=UPDOWN causes hexadecimal print requests to print in hexadecimal updown format; DUMP=ACROSS causes them to print in hexadecimal across format. See “Formats for Displaying or Printing Data” on page 4 for an example of each format.

You can also specify the block size for the output, the number of bytes to skip from the beginning of the object, and the maximum number of bytes to print.

Controlling Print Output

Several SET parameters affect the printed output:

| | |
|-----------------|--|
| DATAHDR | Whether the output contains a data header for each block |
| DBCSPRT | How double-byte characters are printed |
| DUMP | Which format (updown or across) is used for hexadecimal output |
| HEADERPG | Whether the output contains a header page at the beginning |
| PAGESIZE | How many lines per page the output has |
| PAGESKIP | Whether output from each function starts on a new page |
| PRINTLEN | How many columns wide the output is |
| PRINTOUT | Where the output is sent |
| PRTRANS | Whether non-printable characters are translated to blanks |
| RECLIMIT | Which part of each block to print |

For more information about SET parameter values, see “SET (Set Processing Options)” on page 242.

Copying OAM Objects

To copy an OAM object to a different object collection, or to a different name within the same collection, use the **OO** function. You can optionally change the management class and storage class.

Backing up OAM Objects in a Sequential or VSAM Data Set

The following functions let you back up OAM objects:

- OV** Backs up one or more OAM objects to a VSAM data set.
- OS, OQ** Backs up one or more OAM objects to a sequential data set.

Backing up a Single OAM Object

To back up a single OAM object, specify the following:

- The collection name and object name for the object.
- The output data set, according to the normal rules for sequential output or VSAM output functions.
- For VSAM output, whether you want to replace any existing records in the data set.
- Whether you want to create an object header:
 - If you want only the data within the object, specify that you do not want an object header (HEADER=NO). For VSAM, the output data set can be a linear data set, an ESDS or RRDS with an appropriate block size, or a KSDS (if the OAM object contains information that can be used as keys).
 - If you want the object as such (for example, to create a backup copy that you might later copy back to an OAM object collection), you should specify that you want an object header (HEADER=YES). For VSAM, you should use an ESDS.

This object header is intended for use by the SO, QO, and VO functions *only*. It is not designed for use by products other than DITTO.

- The block size for the output data set. The default is 4096.
- The other parameters (*limit*, *volser*, *fromdate*, and *todate*) do not apply when you are backing up a single object.

Backing up a Multiple OAM Objects

To back up multiple OAM objects, specify the following:

- Which objects you want to back up. You must specify a collection name, a volume name, or both. You can back up all of the objects within the collection, volume, or collection/volume combination, or you can back up a subset of the objects based on the following criteria:
 - A generic object name, containing at least one percent sign (%), asterisk (*), or double asterisk (**). (With a non-generic object name, you back up a single object as described previously.)

- The *limit* parameter, to limit the backup to objects on DASD, objects on optical disks, or objects on a specific volser. (If you specify a volser, you must also specify VOLSER for the *limit* parameter.)
- The earliest creation date
- The latest creation date
- The output data set, according to the normal rules for sequential output or VSAM output functions. For VSAM output, the data set must be an ESDS.
- For VSAM output, whether you want to replace any existing records in the data set.
- The block size for the output data set. The default is 4096.
- The *header* parameter does not apply when you are backing up multiple objects. A header is automatically generated for each object. This object header is intended for use by the SO, QO, and VO functions *only*. It is not designed for use by products other than DITTO.

Restoring OAM Objects from a Sequential or VSAM Data Set

The following functions let you restore OAM objects:

VO Restores one or more OAM objects from a VSAM data set.

SO, QO Restores one or more OAM objects from a sequential data set.

To restore one or more objects, you should know whether the data set was created by a single-object backup or a multiple-object backup.

Restoring an Object that was Backed up Individually

Follow the instructions in this section in any of the following cases:

- To restore an object that was backed up as a single object with an object header.
- To restore an object that was backed up as a single object without an object header.
- To create an OAM object from a data set that was not previously an OAM object. (This is equivalent to the previous case, because DITTO does not know if a data set with no header was formerly an OAM object.)

In any of these cases, specify the following:

- The input data set, according to the normal rules for sequential input or VSAM input functions.
- The collection name, object name, storage class, and management class. By default, the values in the object header (if any) are used.
- Whether you want to replace any existing object with the same name in the same collection.

Restoring One or More Objects from a Multiple-Object Backup

Follow the instructions in this section in any of the following cases:

- To restore all of the objects from a multiple-object backup.
- To restore some of the objects from a multiple-object backup.
- To restore one of the objects from a multiple-object backup.

In any of these cases, specify the following:

- The input data set, according to the normal rules for sequential input or VSAM input functions.
- Which object or objects you want to restore. You can restore all of the objects in the data set, or you can restore a subset of the objects based on the following criteria:
 - Collection name
 - Object name or generic object name
 - The *limit* parameter, to restore only objects that were backed up from DASD, from optical disks, or from a specific volser.
 - The earliest creation date
 - The latest creation date
- The storage class and management class.
- Whether you want to replace any existing object with the same name in the same collection.

The object or objects that meet the specified criteria will be restored. Each object will have the same object name and the same collection as it had before it was backed up. You cannot change the object name or collection name with this function.

Erasing OAM Objects

Use the **OE** function to erase an OAM object.

Command Mode Example Using DITTO OAM Functions

Figure 80 on page 141 shows a REXX procedure that erases all objects whose name starts with TEST.

- A** DITTO statement to route print output to the REXX stem variable DITTO..
- B** DITTO statement to print all objects in the collection SYSU.OAM.CLLCT002 whose name begins with TEST.
- C** DITTO statement to erase an object.

```

/* REXX */
/* Delete objects named 'TEST.**' */
/* Route DITTO print output to REXX stem var */
A "DITTO $SET PRINTOUT=REXX"
/* Extract names of 'TEST.**' objects... */
B "DITTO $ODL COLLECTN='SYSU.OAM.CLLCT002',"
    "OBJNAME='TEST.**',FUNCTION=PRINT"
/* Loop over stem and erase the test objects */
do i = 1 to DITTO.0
  Parse var DITTO.i objnam .
  Parse var objnam level '.' .
  if level = 'TEST' then do
C   "DITTO $OE COLLECTN='SYSU.OAM.CLLCT002',"
    "OBJNAME="||objnam||""
    if rc = 0 then say objnam 'erased'
  end
end
end
exit

```

Figure 80. Command Mode Example Using DITTO OAM Functions

Batch Example Using DITTO OAM Functions

Figure 81 shows a DITTO batch job that copies a VSAM data set to an OAM object collection, replacing an existing OAM object. The object is then printed.

- A** DD statement for a VSAM input data set. No DD statements are required for the OAM details.
- B** DITTO control statement to erase an existing OAM object.
- C** DITTO control statement to copy a VSAM data set to an OAM object collection (using the name of the previously erased object). The control statement is split onto two lines.
- D** DITTO control statement to print the new OAM object.

```

//S1      EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
A //TSTESDS DD DSN=DITTO.ESDS.DSET,DISP=SHR
//SYSIN   DD *
B $$DITTO OE COLLECTN=SAMP.OAM.COLLECT,OBJECT=OLD.OAM.OBJ
C $$DITTO VO INPUT=TSTESDS,COLLECTN=SAMP.OAM.COLLECT,
C $$DITTO   OBJECT=OLD.OAM.OBJ
D $$DITTO OP COLLECTN=SAMP.OAM.COLLECT,OBJECT=OLD.OAM.OBJ
$$DITTO EOJ
/*
//

```

Figure 81. Batch Example Using DITTO OAM Functions

Chapter 13. Processing Cards

You can use the following functions to process cards:

Figure 82. DITTO Functions for Processing Cards

| Purpose | Functions | |
|--|---------------|-------------------------|
| Print a listing of a card deck, in character or hexadecimal format | CP | Card Print |
| Copy cards | CL | Card to Library Member |
| | CS, CQ | Card to Sequential Data |
| | CT | Card to Tape |
| | CV | Card to VSAM |
| Duplicate a card deck | CC | Card to Card |
| Interpret a prepunched card deck | II | Interpret Cards |
| Cancel card input, or read through a card input stream without processing it | CCL | Card Close |

Card Input and Output

Functions such as **CT** read input from cards, and functions such as **TC** write output to cards. The actual input source and output destination depend on your environment:

- **MVS**: input is read from `SYSIN`. Output is sent to `SYSPUNCH`.
- **VSE**: input is read from the terminal (full-screen mode) or from `SYSIPT` (any other mode). Output is sent to `SYSPCH`.
- **CMS**: input is read from within a batch job immediately after the `DITTO` card-input command (batch mode), or from the virtual reader address `00C` (any other mode). If a reader file is read completely (to EOF), it is purged. Output is sent to the virtual punch address `00D`.

Functions that read input from cards stop processing when they read a card (or card image) that begins with the delimiter specified in the `SET` parameter `E0D`. (For more information about this parameter, see “`SET` (Set Processing Options)” on page 242.)

Card input records can have a length of 80, a multiple of 80, or 81 (VSE only, where the first column is ignored). Under VSE, `DITTO` can read card images from a system file on tape or disk. In batch mode, card input immediately follows the corresponding control statement.

If you interrupt a card function in line mode under VSE and want to read through and ignore the rest of the input card deck, use the **CCL** (card close) function.

If you have a card punch with the card print feature installed, you can interpret card output. If you are interpreting cards under CMS, the punch device must be attached (not CP spooled).

The **II** function interprets prepunched cards. It reads input from and writes output to a physical punch. Under VSE, `SYSPCH` is used for input and output. Under MVS, input is read from `DITIIPT`, which must be the same physical device as `SYSPUNCH`.

Printing Card Decks

Use the **CP** function to print a card deck in character or hexadecimal format. The `SET` parameter `DUMP=UPDOWN` causes hexadecimal print requests to print in hexadecimal updown format; `DUMP=ACROSS` causes them to print in hexadecimal across format. See “Formats for Displaying or Printing Data” on page 4 for an example of each format.

Controlling Print Output

Several `SET` parameters affect the printed output:

| | |
|-----------------|--|
| DBCSPRT | How double-byte characters are printed |
| DUMP | Which format (updown or across) is used for hexadecimal output |
| HEADERPG | Whether the output contains a header page at the beginning |
| PAGESIZE | How many lines per page the output has |
| PAGESKIP | Whether output from each function starts on a new page |
| PRINTLEN | How many columns wide the output is |
| PRINTOUT | Where the output is sent |

PRTRANS Whether non-printable characters are translated to blanks

RECLIMIT Which part of each record to print

For more information about SET parameter values, see “SET (Set Processing Options)” on page 242.

Copying Card Decks

The following functions let you copy a card deck:

CT Copies a card deck to tape.

CV Copies a card deck to a VSAM file.

CS, CQ MVS or VSE: copies a card deck to a sequential data set on tape, disk, or cards.

CL VSE: copies a card deck to a library member.

CC Copies a card deck to a card punch, and optionally prints the interpreted contents on the cards. The card punch must have the card print feature installed for interpreted output.

II Interprets a prepunched card deck.

Cards

Part 3. DITTO Reference

For users new to DITTO

Part 1, "Introduction to DITTO" on page 1 gives an overview of DITTO functions, and the different ways you can use DITTO.

For users already familiar with DITTO basics

Part 2, "Guide to Using DITTO Functions" on page 29 describes what you can accomplish with DITTO functions. The first two chapters describe browse and edit functions that are available in full-screen mode. The remaining chapters are divided according to the type of input you are working with: catalog entries, tapes, disk records, sequential data (QSAM or SAM), VSAM data, CMS files, VSE library members, OAM objects, or cards.

For experienced DITTO users

Part 3, "DITTO Reference" presents the DITTO functions in alphabetical order.

For special situations

Part 4, "Appendixes" on page 347 contains information on compatibility with previous releases, installing DITTO, controlling DITTO security, and responding to DITTO error messages.

If you are running DITTO in full-screen mode, you can also refer to the online help.

Functions

Chapter 14. DITTO Functions

This reference section contains information about DITTO functions that you can use in line mode, command mode, or batch mode. It also contains some information about functions that you can use in full-screen mode. For more information about full-screen functions, use the Help function key (or command) to see the DITTO online help.

How to Use this Reference Chapter

This chapter lists DITTO functions in alphabetical order. It is intended for readers who already know which function they need, but have questions about the syntax. If you are not sure which function you need, refer to Part 1, "Introduction to DITTO" on page 1 and Part 2, "Guide to Using DITTO Functions" on page 29.

For each function, this chapter contains information under the following headings:

| | |
|-----------------------|--|
| Purpose | A short summary of what you can do with the function. |
| Command Syntax | The order in which you enter parameters in line mode (and command mode without the keyword syntax). |
| Batch Syntax | The keywords and parameters that you enter in batch mode. |
| Keyword Syntax | The keywords and parameters that you enter in command mode. |
| Parameters | <p>A description of each parameter, and (where appropriate) its maximum value and default value.</p> <p>In batch mode, parameters are identified by keywords. In line mode, parameters are identified by the order in which you enter them. In command mode, parameters are identified either by keywords or by the order in which you enter them. In full-screen mode, parameters are identified by text on the panel.</p> <p>If you specify parameters that are not needed for a function, DITTO ignores them. In batch mode, if you specify the same parameter more than once, DITTO uses the first value that you specify.</p> |

How to Read the Command Syntax Diagrams

Command syntax is described using the structure defined below.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

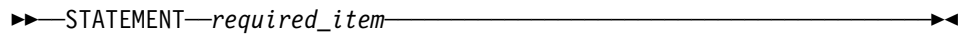
The ►► symbol indicates the beginning of a statement.

The →► symbol indicates that a statement is continued on the next line.

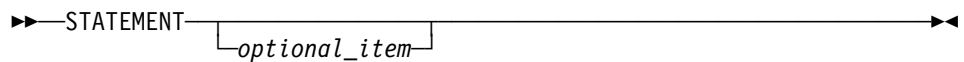
The ► symbol indicates that a statement is continued from the previous line.

The →◄ symbol indicates the end of a statement.

- Required items appear on the horizontal line (the main path).

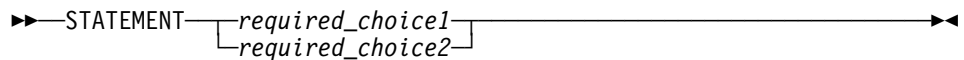


- Optional items appear below the main path.

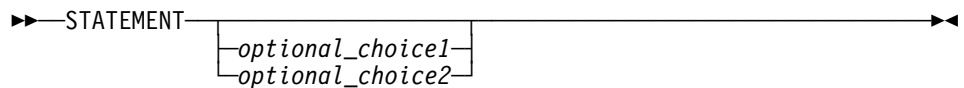


- If you can choose from two or more items, they appear vertically, in a stack.

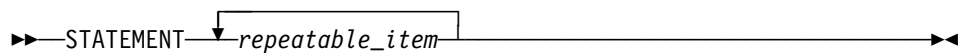
If you *must* choose one of the items, one item of the stack appears on the main path.



If choosing one of the items is optional, the entire stack appears below the main path.



- An arrow returning to the left above the main line indicates an item that can be repeated.



- Variables (for example, *nlrecs*) appear in lowercase letters. They represent user-supplied names or values.
- The command syntax diagram for the SET function also contains the symbols described in “How to Read the Batch Syntax and Keyword Syntax Diagrams” on page 151.

How to Read the Batch Syntax and Keyword Syntax Diagrams

Batch syntax and keyword syntax are described using the structure defined below.

- Keywords (for example, INPUT) appear in uppercase letters. They must be spelled exactly as shown.
- Variables (for example, *n1recs*) appear in lowercase letters. They represent user-supplied names or values.
- Brackets [] indicate that a keyword is optional.
- Braces { } indicate that you must specify one of the keywords or variables within the braces. Within a pair of braces, items are separated by the | symbol, which means “or.”
- In batch mode:
 - Each control statement must begin with \$\$DITTO in columns 1 to 7, followed by a blank in column 8, followed by the function name, followed by the first keyword and parameter (if any).
 - An equal sign (=) must appear between each keyword and its value.
 - Each keyword/value pair, except the last one, must be followed immediately by a comma.
 - To continue a command onto the next line, you do not need a continuation character; simply end the first line with a comma and begin the next line with \$\$DITTO and a blank.
 - Comment statements, which begin with an asterisk (*), are also allowed.
- In command mode (with keyword syntax):
 - Each control statement must begin with DITTO, followed by at least one blank, followed by the command code prefixed by a \$.
 - An equal sign (=) or blank must appear between each keyword and its value.
 - Each keyword/value pair, except the last one, must be followed immediately by a comma or blank.
 - To continue a command onto the next line (within a procedure), follow the normal rules for line continuation in REXX.

APB (Advanced Print Browse)

Purpose Browse Advanced Function Printing (AFP) format data (such as LIST3820 files) at the terminal. This function is available only in full-screen mode.

This function saves the formatted data in a temporary file (under VSE, the file IJSYS01).

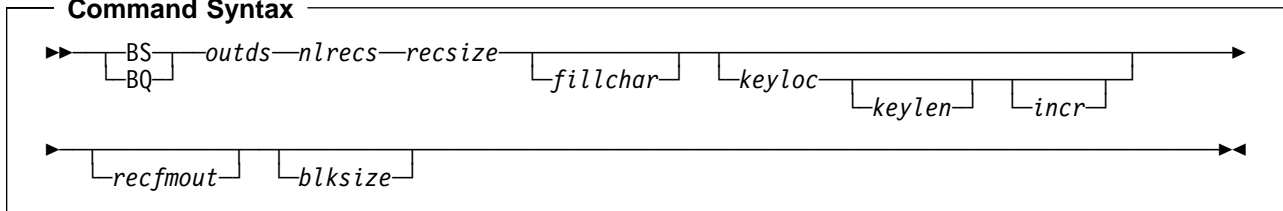
For more information about the APB function, use the online help or see "Browsing Data" on page 32.

BS, BQ (Create Sequential Data) - MVS and VSE only

Purpose Write test data to a sequential data set.

For compatibility with previous releases, you can specify either BS or BQ.

Command Syntax



Batch Syntax

\$\$\$DITTO BS|BQ ...

Keyword Syntax

DITTO \$BS|\$BQ ...

| MVS keywords | VSE keywords | CMS keywords |
|--|---|--------------|
| [{ OUTPUT={QSAMOUT <i>ddname</i> } DSNOUT= <i>dsname</i> [(<i>member</i>)] [VOLSEROUT= <i>volser</i>] [UNITOUT= <i>unit</i>] [DISP={NEW OLD MOD CAT}] [SPACE= <i>space</i>]]] NLRECS= <i>nlrecs</i> RECSIZE= <i>recsize</i> [FILLCHAR={AN BIN RAND <i>fillchar</i> }] [KEYLOC= <i>keyloc</i> [KEYLEN={8 <i>keylen</i> }] [INCR={10 <i>incr</i> }]] [RECFMOUT={U UA UM F FA FM FB FBA FBM FBS FBSA FBSM V VA VM VB VBA VBM VS VSA VSM VBS VBSA VBSM D DA DB DBA DS DSA DBS DBSA}] [BLKSIZE= <i>blksize</i>] | [OUTPUT={ <i>unit</i> SYSnnn <i>vvolser</i> }] [{ FILEOUT={SDSKOUT <i>dlbl</i> } DSNOUT= <i>dsname</i> [VSAMOUT={NO YES}] [BEGIN= <i>begin</i>] [END= <i>end</i>] [UCATOUT= <i>ucat</i>] [DISP={NEW OLD}] [SPACE= <i>space</i>] [AVGSIZE= <i>avgsz</i>]]] NLRECS= <i>nlrecs</i> RECSIZE= <i>recsize</i> [FILLCHAR={AN BIN RAND <i>fillchar</i> }] [KEYLOC= <i>keyloc</i> [KEYLEN={8 <i>keylen</i> }] [INCR={10 <i>incr</i> }]] [RECFMOUT={U F V FB VB}] [BLKSIZE={MAX <i>blksize</i> }] | |

Parameters

- avgsz* The average record size for a new VSAM-managed SAM data set.
- begin* The begin position for a BAM-managed SAM data set in one of the following forms:
 - cylhd* Where the last two digits represent the head, and the other digits represent the cylinder (CKD)
 - Rnnnnnn* To indicate the relative track number (CKD)
 - nnnnnnn* To indicate the physical block number (FBA)

| | |
|-----------------|---|
| <i>blksize</i> | <p>MVS: maximum length of an output block in bytes. Specify it to change the block size of an old data set, or to override any system default for a new data set.</p> <p>VSE: the actual block size for record format FB, otherwise the maximum block size. For record format FB or VB a block size is required, otherwise it is optional. Specify MAX to use the maximum block size allowed for the output device.</p> |
| <i>ddname</i> | Refers to a DD or TSO ALLOC statement. The default is QSAMOUT. |
| <i>disp</i> | <p>Disposition of the output data set.</p> <p>MVS: specify OLD, MOD, NEW, or CAT for NEW,CATLG. The default is NEW.</p> <p>VSE: for VSAM-managed SAM, specify OLD or NEW. The default is NEW.</p> |
| <i>dlbl</i> | Refers to a DLBL statement. The default is SDSKOUT. |
| <i>dsname</i> | Name of a sequential data set. If specified, any MVS DD or VSE DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table. |
| <i>end</i> | <p>The end position for a BAM-managed SAM data set in one of the following forms:</p> <p><i>cylhd</i> Where the last two digits represent the head, and the other digits represent the cylinder (CKD)</p> <p><i>Rnnnnnn</i> To indicate the relative track number (CKD)</p> <p><i>nnnnnnn</i> To indicate the physical block number (FBA)</p> <p><i>Nnnnnnnn</i> To indicate the number of tracks (CKD) or blocks (FBA) to be allocated</p> |
| <i>fillchar</i> | <p>To fill each byte of each record with data, specify one of the following:</p> <p><i>char</i> A character, such as 0 to be written in each byte</p> <p>X'cc' A binary character, such as X'04' to be written in each byte</p> <p>AN To write alphanumeric characters (A to Z and 0 to 9)</p> <p>BIN To write binary characters (X'00' to X'FF')</p> <p>RAND To write random binary characters (X'00' to X'FF')</p> <p>If you specify AN or BIN, characters are written in a “ripple” pattern. For example, if you specify AN, the first 3 records will look like this:</p> <pre> ABC...XYZ0123456789ABC... BC...XYZ0123456789ABC... C...XYZ0123456789ABC... </pre> <p>The default is a blank.</p> |
| <i>incr</i> | Increment value for the sequence field. The default is 10. |
| <i>keylen</i> | Length of the sequence field, from 1 to 9. The default is 8. |
| <i>keyloc</i> | Position (starting with 1) of the sequence field within the output records. If omitted, there is no sequence field. |
| <i>member</i> | For a partitioned data set, the member name. |
| <i>nlnrcs</i> | Number of logical records to be written. The maximum is 99999999. |
| <i>outds</i> | Specification of the output data set. (Command Syntax only.) |
| | <p>MVS:</p> <ul style="list-style-type: none"> • If a JCL <i>ddname</i> or TSO ALLOC statement for QSAMOUT is provided, no input is required. • If the data set is cataloged, specify: <i>dsname[(member)],,{OLD MOD}</i> • If the data set is not cataloged, specify: <i>dsname[(member)],[,unit],{OLD MOD}</i> • For a new data set, specify: <i>dsname,volser[,unit],{NEW CAT MOD},space</i> |

| | |
|-----------------|--|
| | SMS may modify the allocation of new data sets on your system. See your SMS Administrator for additional information. |
| | VSE: |
| | <ul style="list-style-type: none"> • If a DLBL statement for SDSKOUT is provided, no input is required. • For an existing VSAM-managed SAM, specify: <i>dsname</i>,YES,<i>[ucatl]</i>,OLD • For a new VSAM-managed SAM, specify: <i>dsname</i>,YES,<i>[ucatl]</i>,NEW,<i>space</i>,<i>avgsz</i> • For a BAM-managed SAM, specify: <i>dsname</i>,NO,<i>[unit volser]</i>,<i>begin</i>,<i>end</i> |
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <ul style="list-style-type: none"> U Undefined length F Fixed length V Variable length D Variable-length ISO/ANSI tape records B Blocked S Standard format (fixed length) or spanned format (variable length) A Containing ISO/ANSI control characters M Containing machine code control characters |
| | MVS: use <i>recfmout</i> to override existing DCB information. You cannot create a PDS with a record format of FBS. If output data sets with a record format containing the letter D are written to disk, unpredictable results may occur. |
| <i>recsize</i> | Length of each logical record. <p>MVS: the maximum length is 32760.</p> <p>VSE: for a SAM data set in CI format, the maximum is 32761. For any other SAM data set, the maximum is determined by the track size of the output device.</p> |
| <i>space</i> | The number of space units to be allocated for a new data set. Specify <i>n</i> or (<i>n,m</i>), where <i>n</i> is the number of primary units, and <i>m</i> is the number of secondary units. <p>MVS: The space unit is a track.</p> <p>VSE: The space unit is the length of an average record.</p> |
| <i>SYSnnn</i> | Programmer logical unit to which the output device is assigned. |
| <i>ucatl</i> | For VSAM-managed SAM, the data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. <p>VSE: Device number (3 or 4 hexadecimal digits).</p> |
| <i>volser</i> | Volume serial number for a new or non-cataloged data set. |
| <i>vsamout</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the output volume. |

Batch Example (MVS)

```

//BS JOB (acct),'name' Create QSAM Data
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
/* EXAMPLE BS JOB
/* 203 RECORDS ARE WRITTEN TO A QSAM FILE,
/* FBS RECFM. THE OUTPUT FILE CONTAINS
/* RANDOM CHARACTERS. THERE IS A SEQUENCE
/* FIELD 7 CHARACTERS LONG AT POSITION 100.
/*
//DELETE EXEC PGM=IEFBR14
//BSA1 DD DISP=(MOD,DELETE),UNIT=SYSALLDA,
// SPACE=(TRK,(1,1),RLSE),
// DSN=KEISTEW.BS15.A1FBS
/*
//STP1001 EXEC DTOBAT
//QSAMOUT DD DISP=(,CATLG),UNIT=SYSALLDA,
// SPACE=(CYL,(1,1),RLSE),
// RECFM=FBS,DSN=KEISTEW.BS15.A1FBS
//SYSIN DD *
$$DITTO VER
$$DITTO BS OUTPUT=QSAMOUT,NLRECS=203,
$$DITTO RECSIZE=3456,KEYLOC=100,KEYLEN=7,
$$DITTO INCR=25,RECFMOUT=FBS,FILLCHAR=RAND
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB BS - Create SAM Data
/. C
// DLBL BSC36,'DITTO.TESTCASE.TEMP.BS02.CV19',
// 0,VSAM,,CAT=USYSWK2,
// DISP=(NEW,DELETE,DELETE),
// RECORDS=(900,300),RECSIZE=32750
// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO BS FILEOUT=BSC36,NLRECS=6,
$$DITTO RECSIZE=32750,KEYLOC=8,KEYLEN=6,
$$DITTO INCR=24,RECFMOUT=V
$$DITTO EOJ
/*
/&

```

BSF (Backward Space File)

Purpose Move the tape backward one or more tape files. This function moves the tape backward and stops immediately after reading the appropriate number of tape marks. Thus, the tape is positioned *before* a tape mark or at the load point.

Command Syntax

```
▶▶—BSF—intape ┌───┐
                  │   │
                  └───┘ nfiles ▶▶
```

Batch Syntax

```
$$DITTO BSF ...
```

Keyword Syntax

```
DITTO $BSF ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--|--|---|
| [LABEL=BLP] {INPUT= <i>ddname</i> OUTPUT= <i>ddname</i> } [NFILES={1 <i>nfiles</i> }] | {INPUT={ <i>unit</i> SYS <i>nnn</i> } OUTPUT={ <i>unit</i> SYS <i>nnn</i> } } [NFILES={1 <i>nfiles</i> }] | {INPUT= <i>unit</i> OUTPUT= <i>unit</i> } [NFILES={1 <i>nfiles</i> }] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
| the first DITTO function that uses the tape. For BLP processing requirements, see
| "Customizing the Security Environment" in the *DITTO/ESA Installation and Customization*
| *Guide*.

nfiles Number of tape files to be spaced backward. The maximum is 99999; the default is 1. For more information on counting tape files, see "Positioning a Tape" on page 68.

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or VOL1) of the tape volume to be mounted. For a volume set, specify up to five volsers in the form (volser1, volser2...).

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form `[yy]yy.ddd`.

BSR (Backward Space Record)

Purpose Move the tape backward one or more physical records (blocks).

Command Syntax

```
▶—BSR—intape [ nblks ] ◀
```

Batch Syntax

```
$$DITTO BSR ...
```

Keyword Syntax

```
DITTO $BSR ...
```

| MVS keywords | VSE keywords | CMS keywords |
|---|--|--|
| [LABEL=BLP] {INPUT= <i>ddname</i> OUTPUT= <i>ddname</i> } [NBLKS={ <u>1</u> <i>nblks</i> }] | {INPUT={ <i>unit</i> SYS <i>nnn</i> } OUTPUT={ <i>unit</i> SYS <i>nnn</i> }} [NBLKS={ <u>1</u> <i>nblks</i> }] | {INPUT= <i>unit</i> OUTPUT= <i>unit</i> } [NBLKS={ <u>1</u> <i>nblks</i> }] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit*.

LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

nblks Number of tape blocks to be spaced backward. The maximum is 99999999; the default is 1. A tape mark is considered a block.

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or V0L1) of the tape volume to be mounted. For a volume set, specify up to five volsers in the form (volser1, volser2...).

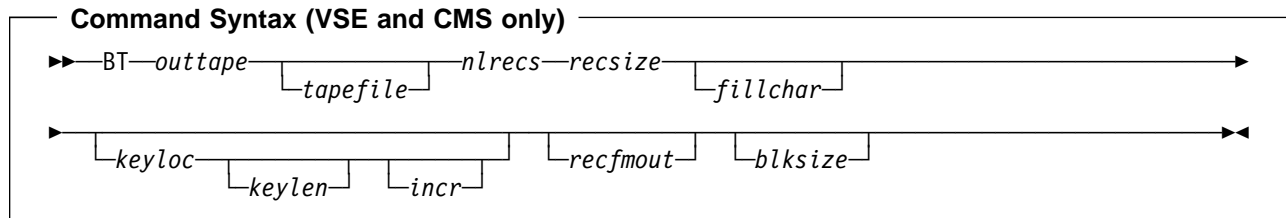
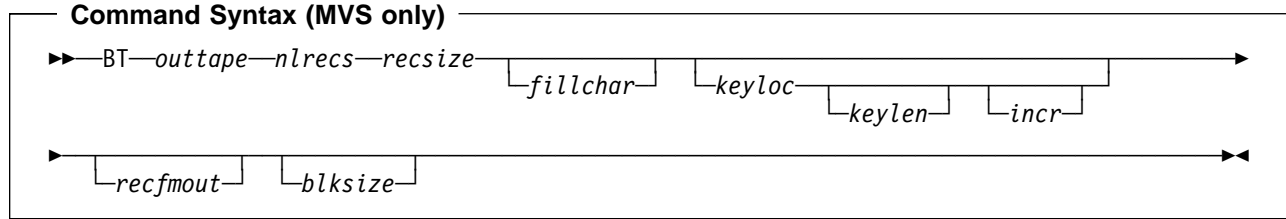
unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form [*yy*]*yy.ddd*.

BT (Create Tape File)

Purpose Write test data to a tape file.



Batch Syntax
`$$DITTO BT ...`

Keyword Syntax
`DITTO $BT ...`

| MVS keywords | VSE keywords | CMS keywords |
|---|--|--|
| <p>[LABEL=BLP] OUTPUT=ddname [DENS=mm] NLRECS=nlrecs RECSIZE=recsize [FILLCHAR={AN BIN RAND fillchar}] [KEYLOC=keyloc [KEYLEN={8 keylen}] [INCR={10 incr}]] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize]</p> | <p>OUTPUT={unit SYSnnn} [DENS=mm] [{ FILEOUT={TAPEOUT tbl OUTFILE=tapefile }] NLRECS=nlrecs RECSIZE=recsize [FILLCHAR={AN BIN RAND fillchar}] [KEYLOC=keyloc [KEYLEN={8 keylen}] [INCR={10 incr}]] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize]</p> | <p>OUTPUT=unit [DENS=mm] [OUTFILE=tapefile] NLRECS=nlrecs RECSIZE=recsize [FILLCHAR={AN BIN RAND fillchar}] [KEYLOC=keyloc [KEYLEN={8 keylen}] [INCR={10 incr}]] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize]</p> |

Parameters

- blksize** If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.
- ddname** Refers to a DD or TSO ALLOC statement. To create a tape file with standard labels, specify AL or SL processing on the DD or TSO ALLOC statement.
- fillchar** To fill each byte of each record with data, specify one of the following:
 - char** A character, such as 0 to be written in each byte
 - X'cc'** A binary character, such as X'04' to be written in each byte
 - AN** To write alphanumeric characters (A to Z and 0 to 9)
 - BIN** To write binary characters (X'00' to X'FF')
 - RAND** To write random binary characters (X'00' to X'FF')

If you specify AN or BIN, characters are written in a “ripple” pattern. For example, if you specify AN, the first 3 records will look like this:

```
ABC...XYZ0123456789ABC...
BC...XYZ0123456789ABC...
C...XYZ0123456789ABC...
```

The default is a blank.

| | |
|------------------|--|
| <i>incr</i> | Increment value for the sequence field. The default is 10. |
| <i>keylen</i> | Length of the sequence field, from 1 to 9. The default is 8. |
| <i>keyloc</i> | Position (starting with 1) of the sequence field within the output records. If omitted, there is no sequence field. |
| LABEL=BLP | Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the <i>DITTO/ESA Installation and Customization Guide</i> . |
| <i>mm</i> | A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67. |
| <i>nlrecs</i> | Number of logical records to be written. The maximum is 99 999 999. |
| <i>outtape</i> | Specification of the output tape unit. (Command Syntax only.) |
| | MVS: <ul style="list-style-type: none"> • If the tape is already allocated specify: <i>ddname[-mm]</i>. • If label processing is desired specify: <i>ddname[-mm],[tapevol],unit,{SL AL},tapedsn,[seqnum],[xdate]</i>. • For NL or BLP tape processing specify: <i>ddname[-mm],[tapevol],unit,{NL BLP},[seqnum],[xdate]</i>. |
| | VSE or CMS: <ul style="list-style-type: none"> • Specify: <i>unit[-mm]</i>. |
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <p>U Undefined length F Fixed length V Variable length D Variable-length ISO/ANSI tape records B Blocked S Spanned format</p> |
| <i>recsize</i> | Length of each logical record. The maximum is 9 999 999. |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| SYSnnn | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapefile</i> | VSE: the file ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed. CMS: the file ID of a tape file, if you want to perform label processing. By default, NL processing is performed. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted, or blank for a non-specific volume request. For a volume set, specify up to five volsers in the form (volser1, volser2...). |

- tlbl* Refers to a TLBL statement to process a labeled tape. The default is TAPEOUT.
- unit* MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
VSE or CMS: Device number (3 or 4 hexadecimal digits).
- xdate* Expiration date in the form [yy]yy.ddd.

Batch Example (MVS)

```
//BT JOB (acct),'name'      Create SL Tape
/*
//DTOBAT  PROC
//DTOBAT  EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.HQGB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//        PEND
/*
/* EXAMPLE BT JOB, SL processing
/* 67 RECORDS ARE WRITTEN TO A TAPE FILE,
/* V RECFM. THE OUTPUT FILE CONTAINS
/* ALPHANUMERIC CHARACTERS. THERE IS A
/* SEQUENCE FIELD AT POSITION 1.
/*
//STP0001  EXEC DTOBAT
//BT1      DD DISP=(NEW,KEEP),UNIT=CART,
//          VOL=(,RETAIN,,,SER=DIT001),
//          LABEL=(1,SL),
//          DSN=KEISTEW.BT05.A1V
//SYSIN    DD *
$$DITTO BT  OUTPUT=BT1,NLRECS=67,RECSIZE=1024,
$$DITTO    KEYLOC=1,RECFMOUT=V,FILLCHAR=AN
$$DITTO EOJ
/*
/*
/* 12 RECORDS ARE WRITTEN TO A TAPE FILE,
/* FB RECFM. THE OUTPUT FILE CONTAINS '*'
/* CHARACTERS. THERE IS A SEQUENCE FIELD
/* AT POSITION 10.
/*
//STP0002  EXEC DTOBAT
//BT2      DD DISP=(NEW,KEEP),UNIT=CART,
//          VOL=(,RETAIN,,,SER=DIT001),
//          LABEL=(2,SL),
//          DSN=KEISTEW.BT05.A2FB
//SYSIN    DD *
$$DITTO BT  OUTPUT=BT2,NLRECS=12,RECSIZE=2048,
$$DITTO    KEYLOC=10,INCR=100,
$$DITTO    RECFMOUT=FB,FILLCHAR='*',
$$DITTO    BLKSIZE=20480
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB BT - Create SL tape file
/. C Note that the output is written to the
/. C 15th file on the tape
// ASSGN SYS010,580
// TLBL BT15,'DITTO.TP.BT02.A15',,DIT001,,15
// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO BT  OUTPUT=SYS010,FILEOUT=BT15,
$$DITTO    NLRECS=24,RECSIZE=12288,
$$DITTO    FILLCHAR=AN,KEYLOC=1,KEYLEN=6,
$$DITTO    INCR=25,RECFMOUT=U
$$DITTO EOJ
/*
/ &
```

Keyword Example (CMS)

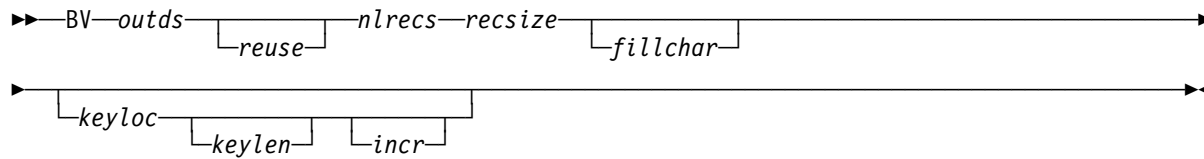
```
/* REXX - create SL tape file */
"DITTO $REW OUTPUT 182" /* goto begin of tape */
"DITTO $BT  OUTPUT 182 OUTFILE TEST.DATA" ,
      "NLRECS 10 RECSIZE 80"
Exit rc
```

BV (Create VSAM File)

Purpose Write test data to a VSAM data set.

CMS: CMS/VSAM is required.

Command Syntax



Batch Syntax

\$\$\$DITTO BV ...

Keyword Syntax

DITTO \$BV ...

| MVS keywords | VSE keywords | CMS keywords |
|--|---|---|
| <pre> [OUTPUT={VSAMOUT ddname} DSNOUT=dsname] [REUSE={NO YES}] NLRECS=nlrecs RECSIZE=recsize [FILLCHAR={AN BIN RAND fillchar}] [KEYLOC=keyloc [KEYLEN={8 keylen}] [INCR={10 incr}]] </pre> | <pre> [FILEOUT={VDSKOUT dlbl} DSNOUT=dsname [UCATOUT=ucat]] [REUSE={NO YES}] [PWDOUT=pwdout] NLRECS=nlrecs RECSIZE=recsize [FILLCHAR={AN BIN RAND fillchar}] [KEYLOC=keyloc [KEYLEN={8 keylen}] [INCR={10 incr}]] </pre> | <pre> DSNOUT=dsname [UCATOUT=ucat] [REUSE={NO YES}] [PWDOUT=pwdout] NLRECS=nlrecs RECSIZE=recsize [FILLCHAR={AN BIN RAND fillchar}] [KEYLOC=keyloc [KEYLEN={8 keylen}] [INCR={10 incr}]] </pre> |

Parameters

ddname Refers to a DD or TSO ALLOC statement. The default is VSAMOUT.

dlbl Refers to a DLBL statement. The default is VDSKOUT.

dsname Name of a VSAM data set.

fillchar To fill each byte of each record with data, specify one of the following:

- char** A character, such as 0 to be written in each byte
- X'cc'** A binary character, such as X'04' to be written in each byte
- AN** To write alphanumeric characters (A to Z and 0 to 9)
- BIN** To write binary characters (X'00' to X'FF')
- RAND** To write random binary characters (X'00' to X'FF')

If you specify AN or BIN, characters are written in a "ripple" pattern. For example, if you specify AN, the first 3 records will look like this:

```

ABC...XYZ0123456789ABC...
BC...XYZ0123456789ABC...
C...XYZ0123456789ABC...
  
```

The default is a blank.

incr Increment value for the sequence field. The default is 10.

keylen Length of the sequence field, from 1 to 9. The default is 8.

keyloc Position (starting with 1) of the sequence field within the output records. If omitted, there is no sequence field.

| | |
|-----------------|--|
| <i>nlrecs</i> | Number of logical records to be written. The maximum is 99 999 999. |
| <i>outds</i> | Specification of the VSAM output data set. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If a DD or TSO ALLOC statement for VSAMOUT is provided, no input is required. • If no DD statement is provided specify: <i>dsname</i>. VSE: <ul style="list-style-type: none"> • If a DLBL statement for VDSKOUT is provided, no input is required. • If no DLBL statement is provided specify: <i>dsname,[ucatl]</i>. CMS: <ul style="list-style-type: none"> • Specify: <i>dsname,[ucatl]</i>. |
| <i>pwdout</i> | Password of a password-protected VSAM data set. |
| <i>reclsize</i> | Length of each logical record. The maximum is 9 999 999. |
| <i>reuse</i> | Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO. |
| <i>ucatl</i> | The data set name of the user catalog. |

Batch Example (MVS)

```
//BV JOB (acct),'name' Create VSAM Data
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//IDCPROC PROC
//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
/* EXAMPLE BV JOB
/* 17 RECORDS ARE WRITTEN TO A NONINDEXED
/* VSAM FILE. THE OUTPUT FILE CONTAINS
/* ALPHANUMERIC CHARACTERS.
/*
//DMBBV02 EXEC IDCPROC,REGION=6M
//SYSIN DD *
DELETE KEISTEW.VSAM.BV25 CLUSTER PURGE ERASE
DEFINE CLUSTER ( -
    NAME( KEISTEW.VSAM.BV25 ) -
    CISZ(4096) -
    FREESPACE(30 30) -
    NONINDEXED -
    KEYS(12 0) -
    RECORDS(1000 100) -
    RECORDSIZE(234 234) -
    REUSE -
    VOL(MVS2W2) ) -
DATA( NAME(KEISTEW.VSAM.BV25.DATA) )
/*
/*
//STP1005 EXEC DTOBAT
//VSAMOUT DD DISP=SHR,DSN=KEISTEW.VSAM.BV25
//SYSIN DD *
$$DITTO VER
$$DITTO BV OUTPUT=VSAMOUT,REUSE=NO,NLRECS=17,
$$DITTO RECSIZE=234,FILLCHAR=AN
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB BV - Create VSAM Data
// EXEC IDCAMS,SIZE=AUTO
DELETE DITTO.TC.TEMP.VSAM.BV2K06 CLUSTER -
PURGE ERASE
DEFINE CLUSTER ( -
    NAME( DITTO.TC.TEMP.VSAM.BV2K06 ) -
    CISZ(4096) -
    FREESPACE(30 30) -
    NONINDEXED -
    RECORDS(1000 100) -
    RECORDSIZE(768 768) -
    REUSE -
    VOL(SYSWK1) ) -
DATA( -
    NAME(DITTO.TC.TEMP.VSAM.BV2K06.DATA) ) -
CATALOG(VSESP.USER.CATALOG)
/*
// DLBL VSAMOUT,'DITTO.TC.TEMP.VSAM.BV2K06',0,
VSAM,,CAT=USYSWK2,
DISP=(OLD,KEEP)

// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO BV FILEOUT=VSAMOUT,REUSE=NO,
$$DITTO NLRECS=367,RECSIZE=768,KEYLOC=1,
$$DITTO KEYLEN=6
$$DITTO EOJ
/*
/ &
```

CC (Card to Card)

Purpose Copy card image records.

Input is read from:

- MVS: SYSIN
- VSE: The terminal in full-screen mode or SYSIPT in any other mode
- CMS: The virtual reader address 00C.

Output is written to:

- MVS: SYSPUNCH
- VSE: SYSPCH
- CMS: The virtual punch address 00D.

| Command Syntax | | |
|-------------------------------|-------------------------------|-------------------------------|
| | | |
| Batch Syntax | Keyword Syntax | |
| \$DITTO CC ... | DITTO \$CC ... | |
| MVS keywords | VSE keywords | CMS keywords |
| [INTERPRET={ <u>NO</u> YES}] | [INTERPRET={ <u>NO</u> YES}] | [INTERPRET={ <u>NO</u> YES}] |

Parameters

interpret Interpret the output. If you specify YES, the output device must be a card punch with the card print feature installed. The default is NO.

CCL (Card Close)—VSE Only

Purpose Cancel card input in line mode. If you interrupt a card-input function and enter CCL, the remainder of the input stream is read and ignored up to and including the end-of-file card. This function is available only on the VSE console.

| Command Syntax |
|----------------|
| |

Parameters: None.

CL (Card to Library Member)—VSE Only

Purpose Copy card image records to a library member. Input is read from the terminal in full-screen mode or from SYSIPT in any other mode.

Command Syntax

```
▶▶ CL outlib member [recfmout] ▶▶
```

Batch Syntax

```
$$DITTO CL ...
```

Keyword Syntax

```
DITTO $CL ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | LIBOUT = <i>lib</i> MEMBEROUT = <i>member</i> [RECFMOUT ={ <i>F</i> <i>S</i> } | |

Parameters

- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- member* Name of the library member in the form *name.type*.
- outlib* Specification of the output library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucat]*.
- recfmout* Record format for the output:
- F** Fixed-length, 80 (the default)
- S** String format (always used if the member type is PHASE).
- ucat* The data set name of the user catalog.
- volser* Volume serial number of the device where the library is located.

Batch Example (VSE)

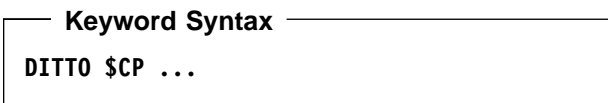
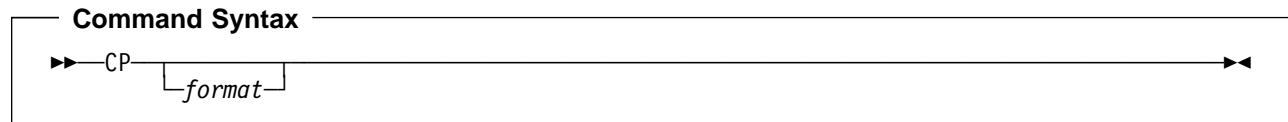
```
// JOB CL - Card to Library Member
// UPSI 1
// EXEC DITTO
$$DITTO CL LIBOUT=PRD2.CONFIG,
$$DITTO MEMBEROUT=VTMAPPL1.B
VTMAPPL1 VBUILD TYPE=APPL
POWER APPL AUTH=(ACQ)
DBDCICS APPL AUTH=(PASS,ACQ)
/*
$$DITTO EOJ
/&
```

CP (Card Print)

Purpose Print card image records in either character or hexadecimal format.

Card input is read from:

- MVS: SYSIN
- VSE: The terminal in full-screen mode or SYSIPT in any other mode
- CMS: The virtual reader address 00C.



| MVS keywords | VSE keywords | CMS keywords |
|------------------------------|------------------------------|------------------------------|
| [FORMAT={ <u>CHAR</u> HEX}] | [FORMAT={ <u>CHAR</u> HEX}] | [FORMAT={ <u>CHAR</u> HEX}] |

Parameters

- format* The format of the output:
- CHAR** Character format (the default)
 - HEX** Hexadecimal format.

CS, CQ (Card to Sequential Data)—MVS and VSE Only

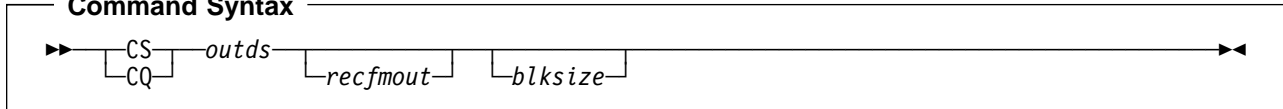
Purpose Copy card image records to a sequential data set.

Card input is read from:

- MVS: SYSIN
- VSE: The terminal in full-screen mode or SYSIPT in any other mode.

For compatibility with previous releases, you can specify either CS or CQ.

Command Syntax



Batch Syntax

\$\$\$DITTO CS|CQ ...

Keyword Syntax

DITTO \$CS|\$CQ ...

| MVS keywords | VSE keywords | CMS keywords |
|---|---|--------------|
| <pre>[{ OUTPUT={QSAMOUT ddname} DSNOUT=dsname[(member)] [VOLSEROUT=volser] [UNITOUT=unit] [DISP={NEW OLD MOD CAT}] [SPACE=space] }] [RECFMOUT={U UA UM F FA FM FB FBA FBM FBS FBSA FBSM V VA VM VB VBA VBM VS VSA VSM VBS VBSA VBSM D DA DB DBA DS DSA DBS DBSA}] [BLKSIZE=blksize]</pre> | <pre>[OUTPUT={unit SYSnnn Vvolser}] [FILEOUT={SDSKOUT dlbl} DSNOUT=dsname [VSAMOUT={NO YES}] [BEGIN=begin] [END=end] [UCATOUT=ucatt] [DISP={NEW OLD}] [SPACE=space] [AVGSIZE=avgsize] }] [RECFMOUT={U F V FB VB}] [BLKSIZE={MAX blksize}]</pre> | |

Parameters

- avgsize* The average record size for a new VSAM-managed SAM data set.
- begin* The begin position for a BAM-managed SAM data set in one of the following forms:
- cyllhd* Where the last two digits represent the head, and the other digits represent the cylinder (CKD)
 - Rnnnnnn* To indicate the relative track number (CKD)
 - nnnnnnn* To indicate the physical block number (FBA)
- blksize* MVS: maximum length of an output block in bytes. Specify it to change the block size of an old data set, or to override any system default for a new data set.
- VSE: the actual block size for record format FB, otherwise the maximum block size. For record format FB or VB a block size is required, otherwise it is optional. Specify MAX to use the maximum block size allowed for the output device.
- ddname* Refers to a DD or TSO ALLOC statement. The default is QSAMOUT.
- disp* Disposition of the output data set.
- MVS: specify OLD, MOD, NEW, or CAT for NEW,CATLG. The default is NEW.
- VSE: for VSAM-managed SAM, specify OLD or NEW. The default is NEW.
- dlbl* Refers to a DLBL statement. The default is SDSKOUT.

| | |
|-----------------|--|
| <i>dsname</i> | Name of a sequential data set. If specified, any MVS DD or VSE DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table. |
| <i>end</i> | The end position for a BAM-managed SAM data set in one of the following forms: <ul style="list-style-type: none"> <i>cylhd</i> Where the last two digits represent the head, and the other digits represent the cylinder (CKD) <i>Rnnnnnn</i> To indicate the relative track number (CKD) <i>nnnnnnn</i> To indicate the physical block number (FBA) <i>Nnnnnnnn</i> To indicate the number of tracks (CKD) or blocks (FBA) to be allocated |
| <i>member</i> | For a partitioned data set, the member name. |
| <i>outds</i> | Specification of the output data set. (Command Syntax only.) |
| | MVS: <ul style="list-style-type: none"> • If a JCL ddname or TSO ALLOC statement for QSAMOUT is provided, no input is required. • If the data set is cataloged, specify: <i>dsname[(member)],,{OLD MOD}</i> • If the data set is not cataloged, specify: <i>dsname[(member)],,[unit],{OLD MOD}</i> • For a new data set, specify: <i>dsname,volser[,unit],{NEW CAT MOD},space</i> <p>SMS may modify the allocation of new data sets on your system. See your SMS Administrator for additional information.</p> <p>VSE: <ul style="list-style-type: none"> • If a DLBL statement for SDSKOUT is provided, no input is required. • For an existing VSAM-managed SAM, specify: <i>dsname,YES,[ucatl],OLD</i> • For a new VSAM-managed SAM, specify: <i>dsname,YES,[ucatl],NEW,space,avgsz</i> • For a BAM-managed SAM, specify: <i>dsname,NO,[unit volser],begin,end</i> </p> |
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <ul style="list-style-type: none"> U Undefined length F Fixed length V Variable length D Variable-length ISO/ANSI tape records B Blocked S Standard format (fixed length) or spanned format (variable length) A Containing ISO/ANSI control characters M Containing machine code control characters <p>MVS: use <i>recfmout</i> to override existing DCB information. You cannot create a PDS with a record format of FBS. If output data sets with a record format containing the letter D are written to disk, unpredictable results may occur.</p> |
| <i>space</i> | The number of space units to be allocated for a new data set. Specify <i>n</i> or <i>(n,m)</i> , where <i>n</i> is the number of primary units, and <i>m</i> is the number of secondary units. <p>MVS: The space unit is a track.</p> <p>VSE: The space unit is the length of an average record.</p> |
| <i>SYSnnn</i> | Programmer logical unit to which the output device is assigned. |
| <i>ucat</i> | For VSAM-managed SAM, the data set name of the user catalog. |

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

VSE: Device number (3 or 4 hexadecimal digits).

volser Volume serial number for a new or non-cataloged data set.

vsamout Specify YES for VSAM-managed SAM. The default is NO.

Vvolser The letter V followed by the serial number of the output volume.

Batch Example (MVS)

```
//CS JOB (acct),'name' Card to QSAM Data
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
/* EXAMPLE CS JOB
/* 3 RECORDS ARE WRITTEN TO THE OUTPUT FILE
/* @@@ IS USED AS THE END-OF-DATA DELIMITER
/* FOR DITTO. AA IS USED AS THE END-OF-DATA
/* DELIMITER FOR SYSIN DATA
/*
//DELETE EXEC PGM=IEFBR14
//QSAMOUT DD DSN=KEISTEW.TEST.CSOUT,
// DISP=(MOD,DELETE),
// SPACE=(TRK,(6,6),RLSE),RECFM=FB,
// LRECL=80,BLKSIZE=8800,
// UNIT=SYSALLDA
/*
//DITTO EXEC DTOBAT
//QSAMOUT DD DSN=KEISTEW.TEST.CSOUT,
// DISP=(,CATLG),
// SPACE=(TRK,(6,6),RLSE),
// RECFM=FB,LRECL=80,
// BLKSIZE=8800
//SYSIN DD DATA,DLM=AA
$$DITTO SET EOD=@@@
$$DITTO CS
// FIRST CARD
// SECOND CARD
/* THE LAST CARD
@@@
$$DITTO EOJ
AA
//
```

Batch Example (VSE)

```
// JOB CS - Card to SAM Data
// UPSI 1
// DLBL SDSKOUT,'SAM.ESDS',,VSAM
// DLBL IJSYSUC,'VSESP.USER.CATALOG',,VSAM
// EXEC DITTO
$$DITTO SET EOD=@@@
$$DITTO CS
// First card
// Second card
/* The last card
@@@
$$DITTO EOJ
/&
```


CT (Card to Tape)

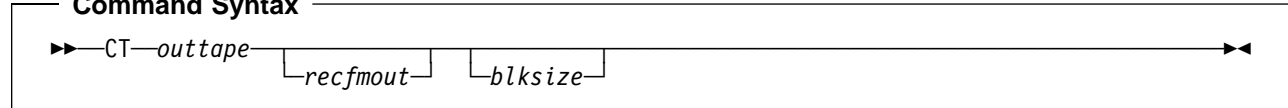
Purpose Copy card image records to tape, starting at the current tape position.

Input is read from:

- MVS: SYSIN
- VSE: The terminal in full-screen mode or SYSIPT in any other mode
- CMS: Virtual reader address 00C.

DITTO writes a tape mark after the last block written on the tape.

Command Syntax



Batch Syntax

\$\$\$DITTO CT ...

Keyword Syntax

DITTO \$CT ...

| MVS keywords | VSE keywords | CMS keywords |
|---|---|--|
| [LABEL=BLP] OUTPUT=ddname [DENS=mm] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] | OUTPUT={unit SYSnnn} [DENS=mm] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] | OUTPUT=unit [DENS=mm] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] |

Parameters

blksize If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.

ddname Refers to a DD or TSO ALLOC statement. To create a tape file with standard labels, specify AL or SL processing on the DD or TSO ALLOC statement.

LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

outtape Specification of the output tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*[-*mm*].
- If label processing is desired specify:
ddname[-*mm*],[*tapevol*],*unit*,{**SL|AL**},*tapedsn*,[*seqnum*],[*xdate*].
- For NL or BLP tape processing specify:
ddname[-*mm*],[*tapevol*],*unit*,{**NL|BLP**},[*seqnum*],[*xdate*].

VSE or CMS:

- Specify: *unit[-mm]*.

| | |
|-----------------|---|
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: U Undefined length F Fixed length V Variable length D Variable-length ISO/ANSI tape records B Blocked S Spanned format |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted, or blank for a non-specific volume request. For a volume set, specify up to five volsers in the form (volser1, volser2...). |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form <i>[yy]yy.ddd</i> . |

CV (Card to VSAM)

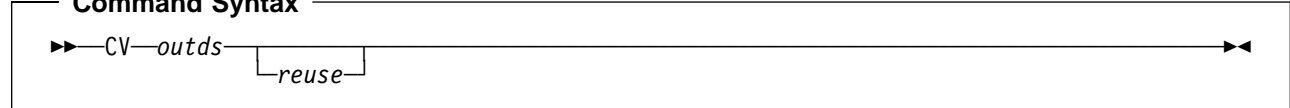
Purpose Copy card image records to a VSAM data set.

Input is read from:

- MVS: SYSIN
- VSE: The terminal in full-screen mode or SYSIPT in any other mode
- CMS: Virtual reader address 00C.

CMS: CMS/VSAM is required.

Command Syntax



Batch Syntax

\$\$DITTO CV ...

Keyword Syntax

DITTO \$CV ...

| MVS keywords | VSE keywords | CMS keywords |
|---|--|---|
| <pre>[{ OUTPUT={VSAMOUT ddname} DSNOUT=dsname }] [REUSE={NO YES}]</pre> | <pre>[{ FILEOUT={VDSKOUT dlbl} DSNOUT=dsname [UCATOUT=ucatl] }] [REUSE={NO YES}] [PWDOUT=pwdout]</pre> | <pre>DSNOUT=dsname [UCATOUT=ucatl] [REUSE={NO YES}] [PWDOUT=pwdout]</pre> |

Parameters

ddname Refers to a DD or TSO ALLOC statement. The default is VSAMOUT.

dlbl Refers to a DLBL statement. The default is VDSKOUT.

dsname Name of a VSAM data set.

outds Specification of the VSAM output data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for VSAMOUT is provided, no input is required.
- If no DD statement is provided specify: *dsname*.

VSE:

- If a DLBL statement for VDSKOUT is provided, no input is required.
- If no DLBL statement is provided specify: *dsname*, [*ucatl*].

CMS:

- Specify: *dsname*, [*ucatl*].

pwdout Password of a password-protected VSAM data set.

reuse Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO.

ucatl The data set name of the user catalog.

Batch Example (MVS)

```

//CV JOB (acct),'name' Card to VSAM
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//IDCPROC PROC
//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
/* EXAMPLE CV JOB
/* 3 RECORDS ARE WRITTEN TO THE OUTPUT VSAM
/* FILE. @@@ IS USED AS THE END-OF-DATA
/* DELIMITER FOR DITTO. AA IS USED AS THE
/* END-OF-DATA DELIMITER FOR SYSIN DATA
/*
//VSAMCR EXEC IDCPROC,REGION=6M
//SYSIN DD *
DELETE KEISTEW.VSAM.CVOUT CLUSTER PURGE -
ERASE
DEFINE CLUSTER ( -
NAME( KEISTEW.VSAM.CVOUT ) -
CISZ(4096) -
FREESPACE(30 30) -
NONINDEXED -
KEYS(12 30) -
RECORDS(1000 100) -
RECORDSIZE(80 80) -
REUSE -
VOL(MVS1W2) ) -
DATA( NAME(KEISTEW.VSAM.CVOUT.DATA) )
/*
/*
//DITTO EXEC DTOBAT
//VSAMOUT DD DSN=KEISTEW.VSAM.CVOUT,DISP=SHR
//SYSIN DD DATA,DLM=AA
$$DITTO SET EOD=@@@
$$DITTO CV REUSE=YES
// FIRST CARD
// SECOND CARD
/* THE LAST CARD
@@@
$$DITTO EOJ
AA
//

```

Batch Example (VSE)

```

// JOB CV - Card to VSAM
// EXEC IDCAMS,SIZE=AUTO
DELETE DITTO.TC.TEMP.VSAM.CV2G43 CLUSTER -
PURGE ERASE
DEFINE CLUSTER ( -
NAME( DITTO.TC.TEMP.VSAM.CV2G43 ) -
CISZ(4096) -
FREESPACE(30 30) -
NONINDEXED -
RECORDS(1000 100) -
RECORDSIZE(80 80) -
REUSE -
VOL(SYSWK1) ) -
DATA( -
NAME(DITTO.TC.TEMP.VSAM.CV2G43.DATA) ) -
CATALOG(VSESP.USER.CATALOG)
/*
// DLBL VDSKOUT,'DITTO.TC.TEMP.VSAM.CV2G43',
0,VSAM,,CAT=USYSWK2,
DISP=(OLD,KEEP)

// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO SET EOD=@@@
$$DITTO CV REUSE=YES
// First card
// Second card
/* The last card
@@@
$$DITTO EOJ
/*
/&

```

DB (Disk Browse)

Purpose Browse disk records at the terminal. This function is available only in full-screen mode.

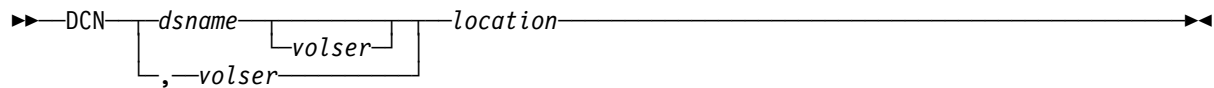
For more information about the DB function, use the online help or see “Browsing Data” on page 32. For information about using this function with disk fullpack processing under MVS, see “MVS: Using Disk Fullpack Processing” on page 89.

DCN (Disk to Console)

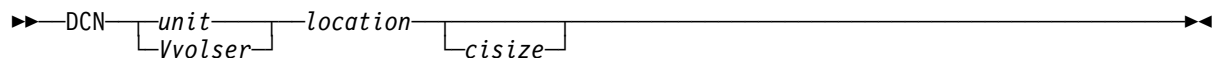
Purpose Display all or part of a disk record at the console, in character or hexadecimal format.

This function is available only in line mode. If you enter DCN in full-screen mode, the DB function is invoked.

Command Syntax (MVS only)



Command Syntax (VSE and CMS only)



Parameters

cisize For an FBA disk, the control interval size. Specify a control interval size to work with only the data portion of a control interval. Omit the control interval size (or specify 0) to work with data as it is physically stored in the FBA blocks.

dsname Name of a disk data set. For fullpack processing (as described in "MVS: Using Disk Fullpack Processing" on page 89), do not specify a *dsname*.

location Disk record address in one of the following forms:

cyl-hd-rrr The cylinder, head and record number (CKD)

Rnnnnnn-rrr The track and record number (CKD)

nnnnnnn The physical block number (FBA)

rrr can be 0 to display record zero information or *ha* for home address information.

MVS: depending on the value of the SET parameter CYLHD, you specify either the absolute disk location or the relative location within the data set.

unit Device number (3 or 4 hexadecimal digits).

volser Volume serial number for a non-cataloged data set or for fullpack processing.

Vvolser The letter V followed by the serial number of the disk volume.

Subcommands

- CNT** Displays the RECORD COUNT field data for a CKD device
- D,*n*** Displays data in character format, starting with byte *n* if specified
- DX,*n*** Displays data in hexadecimal updown format, starting with byte *n* if specified
- K,*n*** Displays the CKD key in character format, starting with byte *n* if specified
- KX,*n*** Displays the CKD key in hexadecimal updown format, starting with byte *n* if specified
- P** Prints the current record in character format
- PX** Prints the current record in hexadecimal updown format
- B** Selects the previous record for subsequent display or printing
- N** Selects the next record for subsequent display or printing
- location* Selects a new record address for subsequent display or printing
- ?** Gives help information
- QUIT** Ends the DCN function.

DID (Alter Disk VOLID)—VSE and CMS Only

Purpose Display or change the existing disk volume identifier.

This function is not available in batch mode.

Command Syntax

```
▶—DID—unit—Vvolser—newvolser—▶
```

Keyword Syntax

DITTO \$DID ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|---|
| | INPUT={ <i>unit</i> SYS <i>nnn</i> <i>Vvolser</i> } [NEWVOL= <i>newvolser</i>] | INPUT={ <i>unit</i> <i>Vvolser</i> } [NEWVOL= <i>newvolser</i>] |

Parameters

newvolser New volume identifier. Omit this if you only want to display the volume identifier.

SYS*nnn* Programmer logical unit to which the disk device is assigned.

unit Device number (3 or 4 hexadecimal digits).

Vvolser The letter V followed by the serial number of the disk volume.

Keyword Example (CMS)

```
/* REXX - Alter Disk VOLID */
"DITTO $DID INPUT SCRTH1 NEWVOL SCRTH2"
Exit rc
```

DLA (Display Label Area)—VSE Only

Purpose Display the contents of the VSE label area.

In full-screen mode, you can print or display the list; if you display the list, you can access any of the listed data sets. The commands available to work with the data sets depend on the type of data (catalog, library, VSAM, or BAM).

The information can be sorted by data set name or label name.

For more information about using this function in full-screen mode, see “VSE: Displaying the Contents of the Label Area” on page 92 and the online help.

Command Syntax

```
DLA [subarea] [templbl] [sortby]
```

Batch Syntax

```
$$DITTO DLA ...
```

Keyword Syntax

```
DITTO $DLA ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | [SUBAREA={SYSTEM ALL part class}] [TEMPLBL={NO YES}] [SORTBY={LABEL NAME VOLSER}] | |

Parameters

sortby One of the following:

LABEL Sorts by label name (the default)
NAME Sorts by data set name
VOLSER Sorts by volume serial number.

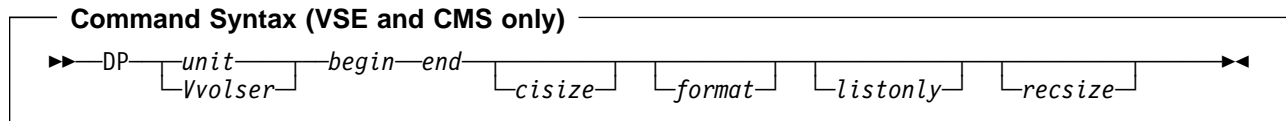
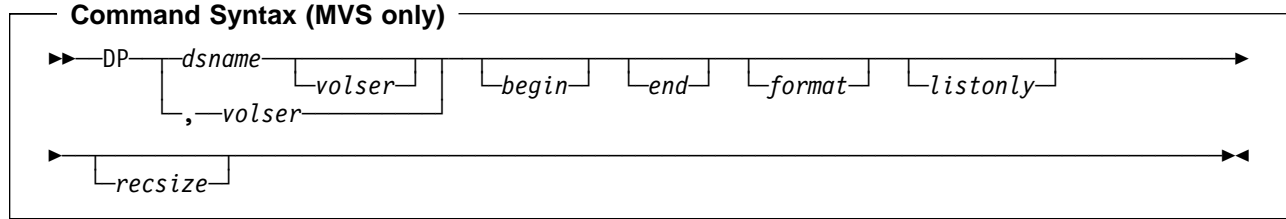
subarea One of the following:

SYSTEM Show label information from the system standard subarea (the default)
ALL Show label information from all subareas
part Show label information from the specified partition
class Show label information with the specified class

templbl Specify YES to include temporary labels (user labels). By default, only permanent labels (standard labels) are included.

DP (Disk Print)

Purpose Print physical disk records in character or hexadecimal format.



Batch Syntax
 \$\$DITTO DP ...

Keyword Syntax
 DITTO \$DP ...

| MVS keywords | VSE keywords | CMS keywords |
|---|--|--|
| { INPUT= <i>ddname</i> DSNAME= <i>dsname</i> [VOLSER= <i>volser</i>] VOLSER= <i>volser</i> } [BEGIN= <i>begin</i>] [END= <i>end</i>] [FORMAT={CHAR HEX}] [LISTONLY={KD CNT KEY CKD}] [RECSIZE= <i>recsize</i>] | INPUT={ <i>unit</i> SYS <i>nnn</i> <i>vvolser</i> } BEGIN= <i>begin</i> END= <i>end</i> [CISIZE= <i>cisize</i>] [FORMAT={CHAR HEX}] [LISTONLY={KD CNT KEY CKD}] [RECSIZE= <i>recsize</i>] | INPUT={ <i>unit</i> <i>vvolser</i> } BEGIN= <i>begin</i> END= <i>end</i> [CISIZE= <i>cisize</i>] [FORMAT={CHAR HEX}] [LISTONLY={KD CNT KEY CKD}] [RECSIZE= <i>recsize</i>] |

Parameters

begin Start position in one of the following forms:

- cylhd* Where the last two digits represent the head, and the other digits represent the cylinder (CKD)
- Rnnnnnn* To indicate the track number (CKD)
- nnnnnnn* To indicate the physical block number (FBA)
- ** The first track of the data set (MVS only, the default).

MVS: depending on the value of the SET parameter CYLHD, you specify either the absolute disk location or the relative location within the data set.

cisize This parameter is required only when processing FBA disks. When working with CKD disks, omit the CISIZE= keyword (keyword syntax), or omit the comma for the parameter (command syntax).

For an FBA disk, the control interval size. Specify a control interval size to work with only the data portion of a control interval. Omit the control interval size (or specify 0) to work with data as it is physically stored in the FBA blocks.

ddname Refers to a DD or TSO ALLOC statement.

dsname Name of a disk data set. For fullpack processing (as described in "MVS: Using Disk Fullpack Processing" on page 89), do not specify a *dsname*.

end End position in one of the following forms:

- cylhd* Where the last two digits represent the head, and the other digits represent the cylinder (CKD)

DP

Rnnnnnn To indicate the track number (CKD)
nnnnnnn To indicate the physical block number (FBA)
Nnnnnnnn The number of tracks (CKD) or blocks (FBA) to be processed
* The last track of the data set (MVS only, the default).

MVS: depending on the value of the SET parameter CYLHD, you specify either the absolute disk location or the relative location within the data set.

format The format of the output:

CHAR Character format (the default)
HEX Hexadecimal format.

listonly For a CKD disk, the information you want printed:

KD Key and data fields of each record (the default)
CNT Home address and count fields of each record
CKD Home address and record 0 fields, followed by the count, key, and data fields
KEY Only the key fields.

recsize Logical record size to be used to deblock each physical record into logical records. The maximum is 65535. The physical block size does not need to be a multiple of *recsize*. For a CKD disk, only the data field is deblocked.

SYSnnn Programmer logical unit to which the disk device is assigned.

unit Device number (3 or 4 hexadecimal digits).

volser Volume serial number for a non-cataloged data set or for fullpack processing.

Vvolser The letter V followed by the serial number of the disk volume.

Batch Example (MVS)

```
//DP JOB (acct),'name' Disk Print
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DITTO EXEC DTOBAT
//DISK DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSIN DD *
$$DITTO SET CYLHD=RELATIVE
$$DITTO DP INPUT=DISK,BEGIN=R0,END=N10
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB DP - Disk Print
// UPSI 1
// EXEC DITTO
$$DITTO DP INPUT=VSYSWK1,BEGIN=R0,END=N10
$$DITTO EOJ
/&
```

DRL (Disk Record Load)

Purpose Alter a physical record on a disk.

This function is not available in batch mode or command mode. In full-screen mode, DU is usually more convenient than DRL.

Command Syntax (MVS only)

```

▶ DRL dsname [volser] location
      [, volser]
  
```

Command Syntax (VSE and CMS only)

```

▶ DRL [unit] [volser] location [cisize]
  
```

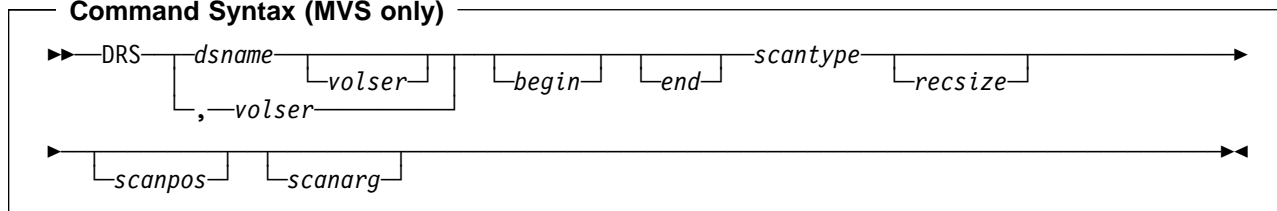
Parameters

- cisize* For an FBA disk, the control interval size. Specify a control interval size to work with only the data portion of a control interval. Omit the control interval size (or specify 0) to work with data as it is physically stored in the FBA blocks.
- dsname* Name of a disk data set. For fullpack processing (as described in "MVS: Using Disk Fullpack Processing" on page 89), do not specify a *dsname*.
- location* Disk record address in one of the following forms:
- cyl-hd-rrr* The cylinder, head and record number (CKD)
 - Rnnnnnn-rrr* The track and record number (CKD)
 - nnnnnnn* The physical block number (FBA)
- MVS: depending on the value of the SET parameter CYLHD, you specify either the absolute disk location or the relative location within the data set.
- unit* Device number (3 or 4 hexadecimal digits).
- volser* Volume serial number for a non-cataloged data set or for fullpack processing.
- Vvolser* The letter V followed by the serial number of the disk volume.

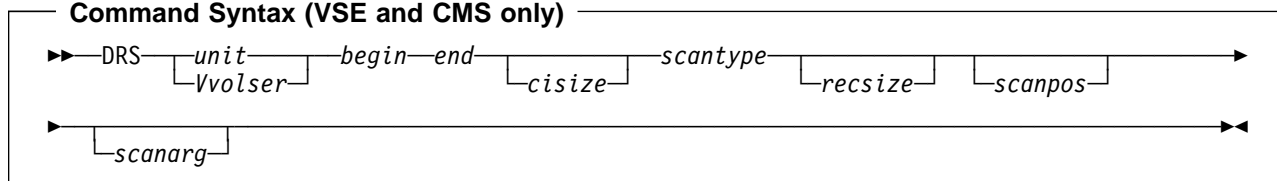
DRS (Disk Record Scan)

Purpose Locate data on a disk.

Command Syntax (MVS only)



Command Syntax (VSE and CMS only)



Batch Syntax

\$\$DITTO DRS ...

Keyword Syntax

DITTO \$DRS ...

| MVS keywords | VSE keywords | CMS keywords |
|---|--|---|
| { INPUT=ddname DSNAME=dsname [VOLSER=volser] VOLSER=volser } [BEGIN=begin] [END=end] SCANTYPE={K KP D DP E} [RECSIZE=recsize] [SCANPOS={1 scanpos}] [SCANARG=scanarg] | INPUT={unit SYSnnn Vvolser} BEGIN=begin END=end [CISIZE=csize] SCANTYPE={K KP D DP E} [RECSIZE=recsize] [SCANPOS={1 scanpos}] [SCANARG=scanarg] | INPUT={unit Vvolser} BEGIN=begin END=end [CISIZE=csize] SCANTYPE={K KP D DP E} [RECSIZE=recsize] [SCANPOS={1 scanpos}] [SCANARG=scanarg] |

Parameters

begin Start position in one of the following forms:

cylhd Where the last two digits represent the head, and the other digits represent the cylinder (CKD)

Rnnnnnn To indicate the track number (CKD)

nnnnnnn To indicate the physical block number (FBA)

*** The first track of the data set (MVS only, the default).

MVS: depending on the value of the SET parameter CYLHD, you specify either the absolute disk location or the relative location within the data set.

csize This parameter is required only when processing FBA disks. When working with CKD disks, omit the CISIZE= keyword (keyword syntax), or omit the comma for the parameter (command syntax).

For an FBA disk, the control interval size. Specify a control interval size to work with only the data portion of a control interval. Omit the control interval size (or specify 0) to work with data as it is physically stored in the FBA blocks.

ddname Refers to a DD or TSO ALLOC statement.

dsname Name of a disk data set. For fullpack processing (as described in "MVS: Using Disk Fullpack Processing" on page 89), do not specify a *dsname*.

| | |
|-----------------|--|
| <i>end</i> | End position in one of the following forms: <i>cylhd</i> Where the last two digits represent the head, and the other digits represent the cylinder (CKD) <i>Rnnnnnn</i> To indicate the track number (CKD) <i>nnnnnnn</i> To indicate the physical block number (FBA) <i>Nnnnnnn</i> The number of tracks (CKD) or blocks (FBA) to be processed * The last track of the data set (MVS only, the default). MVS: depending on the value of the SET parameter <i>CYLHD</i> , you specify either the absolute disk location or the relative location within the data set. |
| <i>recsize</i> | Logical record size to be used to deblock each physical record into logical records. The maximum is 65 535. The physical block size does not need to be a multiple of <i>recsize</i> . |
| <i>scanarg</i> | The string to be searched for. The string can contain up to 50 characters and can be: <ul style="list-style-type: none"> • A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored. • A character string enclosed in single quotation marks. The string <i>can</i> contain blanks and commas; the string must match exactly (including case). • X followed by a hexadecimal string enclosed in single quotation marks, for example, X'C1C2C3'. The <i>scanarg</i> parameter is required unless <i>scantype</i> is E. |
| <i>scanpos</i> | Offset (byte position) where the search begins within a record. The default is 1. |
| <i>scantype</i> | One of the following: <ul style="list-style-type: none"> D Scans data fields <i>only</i> at the position in the record specified by <i>scanpos</i> DP Scans data fields <i>starting</i> at the position in the record specified by <i>scanpos</i> K Scans key fields <i>only</i> at the position in the record specified by <i>scanpos</i> (CKD disks only) KP Scans key fields <i>starting</i> at the position in the record specified by <i>scanpos</i> (CKD only) E Scans for end-of-file records. |
| <i>SYSnnn</i> | Programmer logical unit to which the disk device is assigned. |
| <i>unit</i> | Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number for a non-cataloged data set or for fullpack processing. |
| <i>Vvolser</i> | The letter V followed by the serial number of the disk volume. |

Batch Example (MVS)

```

//DRS JOB (acct),'name' Disk Record Scan
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DELETE EXEC PGM=IEFBR14
//SYSUT2 DD DSN=KEISTEW.TEMP.SEARCH.FILE,
// DISP=(MOD,DELETE),
// UNIT=SYSALLDA,
// SPACE=(TRK,(3,1),RLSE),
// RECFM=FB,LRECL=80,BLKSIZE=24000
//*
//MAKEFILE EXEC PGM=IEBGENER
//SYSIN DD *
/*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD *
Andrew Astle
Tyrone Dalais
Grant Sutherland
Keith Stewart
Graham Hannington
John Leverington
Bob McCormack
Liz Rushton
/*
//SYSUT2 DD DSN=KEISTEW.TEMP.SEARCH.FILE,
// DISP=(,CATLG),
// UNIT=SYSALLDA,
// SPACE=(TRK,(3,1),RLSE),
// RECFM=FB,LRECL=80,BLKSIZE=24000
//*
//DITTO EXEC DTOBAT
//SYSPRINT DD SYSOUT=*
//DISK DD DSN=KEISTEW.TEMP.SEARCH.FILE,
// DISP=SHR
//SYSIN DD *
$$DITTO DRS INPUT=DISK,BEGIN=*,END=*,
$$DITTO SCANTYPE=DP,SCANARG='Leverington'
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB DRS - Disk Record Scan
// UPSI 1
// EXEC DITTO
$$DITTO DRS INPUT=VSYSWK1,BEGIN=R0,END=N100,
$$DITTO SCANTYPE=DP,SCANARG='Perth'
$$DITTO EOJ
/&

```

DSE (Data Set Extents)

Purpose Display the begin and end of the extents of a data set. This function can only be used on OS (MVS) or DOS (VSE) formatted disks.

CMS: CMS/DOS is required.

Command Syntax (MVS only)

```
▶ DSE dsname [volser]
      ,volser
```

Command Syntax (VSE and CMS only)

```
▶ DSE dsname [unit]
      Vvolser
```

Batch Syntax

```
$$DITTO DSE ...
```

Keyword Syntax

```
DITTO $DSE ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--|---|--------------------------------------|
| { INPUT=ddname DSNAME=dsname [VOLSER=volser] VOLSER=volser } | DSNIN=dsname INPUT={unit SYSnnn Vvolser} | DSNIN=dsname INPUT={unit Vvolser} |

Parameters

- ddname* Refers to a DD or TSO ALLOC statement.
- dsname* Name of a disk data set. For fullpack processing (as described in “MVS: Using Disk Fullpack Processing” on page 89), do not specify a *dsname*.
- SYSnnn* Programmer logical unit to which the disk device is assigned.
- unit* Device number (3 or 4 hexadecimal digits).
- volser* Volume serial number for a non-cataloged data set or for fullpack processing.
- Vvolser* The letter V followed by the serial number of the disk volume.

Batch Example (MVS)

```
//DP JOB (acct),'name' Data Set Extents
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//
//*
//DITTO EXEC DTOBAT
//DISK DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSIN DD *
$$DITTO DSE INPUT=DISK
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB DP - Data Set Extents
// UPSI 1
// EXEC DITTO
$$DITTO DSE INPUT=VSYSWK1,DSNIN=DTT.ITT.SAMDS
$$DITTO EOJ
/ &
```

DTE (Disk Track Edit)

Purpose Browse and update disk tracks at the terminal. This function is available only in full-screen mode. For more information about the DTE function, use the online help or see Chapter 4, “Editing Data in Full-Screen Mode” on page 39. For information about using this function with disk fullpack processing under MVS, see “MVS: Using Disk Fullpack Processing” on page 89.

DU (Disk Update)

Purpose Browse and update disk records at the terminal. This function is available only in full-screen mode. For more information about the DU function, use the online help or see “Updating Disk Records in Full-Screen Mode” on page 93. For information about using this function with disk fullpack processing under MVS, see “MVS: Using Disk Fullpack Processing” on page 89.

DUMP (Dump CMS Files to Tape)—CMS Only

Purpose High performance copy of files to tape for backup.

In command mode, the following return codes are set:

- 0 Successful completion
- 4 File(s) not found
- 16 Severe error (indicated by messages).

Command Syntax

►—DUMP—*infile*—*outtape*—◄

Batch Syntax

\$DITTO DUMP ...

Keyword Syntax

DITTO \$DUMP ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--------------|--|
| | | INFILE= <i>infile</i> OUTPUT= <i>unit</i> [DENS= <i>mm</i>] |

Parameters

- infile* Specifies the CMS input file identifier: *filename filetype [filemode]*, enclosed in single quotes. Within the filename or filetype, you can include a percent sign (%) to represent exactly one character, or an asterisk (*) to represent any number of characters (or no characters). The default filemode is A.
- mm* A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.
- outtape* Specification of the output tape unit. (Command Syntax only.) Specify: *unit*[*-mm*].
- unit* Device number (3 or 4 hexadecimal digits).

Keyword Example (CMS)

```
/* REXX - DUMP CMS File to Tape */
"DITTO $DUMP INFILE '* * A' OUTPUT 181"
Exit rc
```

DVB (Data In Virtual Browse)—MVS Only

Purpose Browse a data-in-virtual data set at the terminal. This function is available only in full-screen mode under MVS. For more information about the DVB function, use the online help or see “Browsing Data” on page 32.

DVT (Display VTOC)

Purpose Display a list of the data sets on a disk volume (obtained from the disk VTOC).

In full-screen mode, you can print or display the list; if you display the list, you can browse, print, rename, scratch or update any of the listed files. In command mode under CMS or MVS, you can invoke the full-screen mode DVT function. In all other modes, you can only print the list.

The information can be sorted by data set name, extent, or date of creation or expiration. The listing also shows free extents on the disk.

Note: DVT shows “never to expire” expiration dates of 99/999 and 99/366 as 9999.999 and 9999.366, respectively. An expiration date that was explicitly specified as 99/365 is also considered “never to expire,” and is shown as 9999.365. However, an expiration date of 99/365 that was calculated from a specified retention period (current date + number of days = 99/365) *will* expire, and is shown as 1999.365. If you sort the list by expiration date, values beginning with 9999 appear first.

For more information about using this function in full-screen mode, see “Using the DVT Function in Full-Screen Mode” on page 90.

This function can only be used on OS (MVS) or DOS (VSE) formatted disks.

CMS: CMS/DOS is required.

Command Syntax (MVS only)

```
▶ DVT volser [dsnmask] [sortby] [function] ▶▶
```

Command Syntax (VSE only)

```
▶ DVT [unit] [vvolser] [dsnmask] [sortby] ▶▶
```

Command Syntax (CMS only)

```
▶ DVT [unit] [vvolser] [dsnmask] [sortby] [function] ▶▶
```

Batch Syntax

```
$$DITTO DVT ...
```

Keyword Syntax

```
DITTO $DVT ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--|---|---|
| VOLSER =volser [DSNAME] ={** dsnmask} [SORTBY] ={NAME EXTENT DATE EDATE} [FUNCTION] ={LIST PRINT} | INPUT ={unit SYSnnn vvolser} [DSNAME] ={** dsnmask} [SORTBY] ={NAME EXTENT DATE EDATE} | INPUT ={unit vvolser} [DSNAME] ={** dsnmask} [SORTBY] ={NAME EXTENT DATE EDATE} [FUNCTION] ={LIST PRINT} |

Parameters

dsnmask A generic data set name, or search pattern. Within the data set name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers. The default is **.

Under MVS in any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes. For example, the default of ** lists all entries in batch mode; in any other mode, it lists all entries that have the user's TSO prefix as high-level qualifier.

function The command that you want to execute:

LIST Display the list (the default)

PRINT Print the list.

sortby One of the following:

NAME Sorts by data set name (the default)

DATE Sorts by creation date

EDATE Sorts by expiration date

EXTENT Sorts by disk extent.

SYSnnn Programmer logical unit to which the disk device is assigned.

unit Device number (3 or 4 hexadecimal digits).

volser Volume serial number of the disk.

Vvolser The letter V followed by the serial number of the disk volume.

Batch Example (MVS)

```

//DVT JOB (acct),'name' Display VTOC
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//
//*
//DITTO EXEC DTOBAT
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$DITTO DVT VOLSER=SCPMV5,DSNAME=**,SORTBY=DATE
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB DVT - Display VTOC
// UPSI 1
// EXEC DITTO
$$DITTO DVT INPUT=VSYSWK1,DSNAME=**,SORTBY=DATE
$$DITTO EOJ
/*
/&

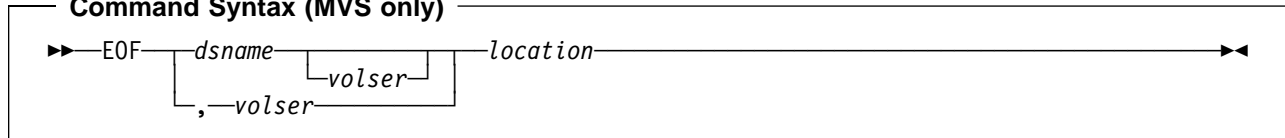
```

EOF (Write EOF Record)

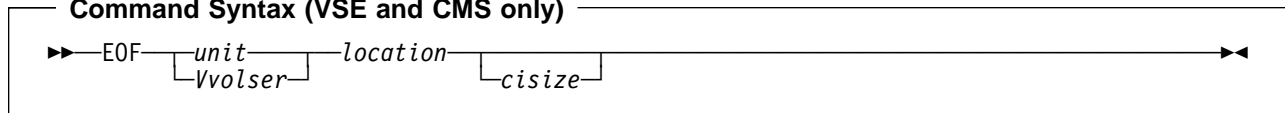
Purpose Retrieve the requested disk record and rewrite it as a software end-of-file (EOF) indicator. For a CKD disk, the EOF record has a data length of zero; for an FBA disk, a control interval consisting entirely of zeros is written.

This function is not available in batch mode or command mode.

Command Syntax (MVS only)



Command Syntax (VSE and CMS only)



Parameters

csize For an FBA disk, the control interval size. DITTO writes a control interval that contains all zeros, so you must provide accurate control interval information. The disk address must denote the first block of an existing control interval.

dsname Name of a disk data set. For fullpack processing (as described in "MVS: Using Disk Fullpack Processing" on page 89), do not specify a *dsname*.

location Disk record address in one of the following forms:

- cyl-hd-rrr* The cylinder, head and record number (CKD)
- Rnnnnnn-rrr* The track and record number (CKD)
- nnnnnnn* The physical block number (FBA)

MVS: depending on the value of the SET parameter CYLHD, you specify either the absolute disk location or the relative location within the data set.

unit Device number (3 or 4 hexadecimal digits).

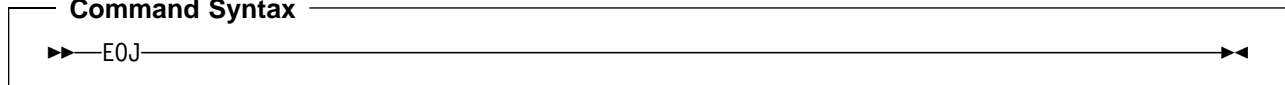
volser Volume serial number for a non-cataloged data set or for fullpack processing.

Vvolser The letter V followed by the serial number of the disk volume.

EOJ (End of Job)

Purpose Exit DITTO.

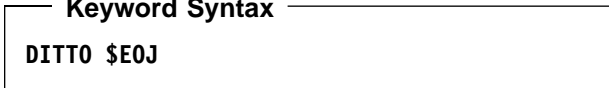
Command Syntax



Batch Syntax



Keyword Syntax



Parameters: None.

ERT (Erase Tape)

Purpose Erase a tape from the current position until the end of the tape. This function writes two tape marks at the current tape position, erases the remainder of the tape, then rewinds the tape.

ERT can be used only with tape units that support the data security erase I/O command code, such as the IBM 3400 series.

Command Syntax

▶—ERT—*outtape*—◀

Batch Syntax

\$\$DITTO ERT ...

Keyword Syntax

DITTO \$ERT ...

| MVS keywords | VSE keywords | CMS keywords |
|--|---|---|
| [LABEL=BLP] OUTPUT= <i>ddname</i> [DENS= <i>mm</i>] | OUTPUT={ <i>unit</i> SYS <i>nnn</i> } [DENS= <i>mm</i>] | OUTPUT= <i>unit</i> [DENS= <i>mm</i>] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

outtape Specification of the output tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*[-*mm*].
- If label processing is desired specify:
ddname[-*mm*],[*tapevol*],*unit*,{SL|AL},*tapedsn*,[*seqnum*],[*xdate*].
- For NL or BLP tape processing specify:
ddname[-*mm*],[*tapevol*],*unit*,{NL|BLP},[*seqnum*],[*xdate*].

VSE or CMS:

- Specify: *unit*[-*mm*].

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or V0L1) of the tape volume to be mounted, or blank for a non-specific volume request.

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form [*yy*]*yy*.*ddd*.

FB (CMS File Browse)—CMS Only

Purpose Browse a CMS file at the terminal. This function is available only in full-screen mode under CMS. For more information about the FB function, use the online help or see “Browsing Data” on page 32.

FL (CMS File to Library Member)—CMS/VSE Mode Only

Purpose Copy records from a CMS file to a library member. This function is available only in full-screen mode VSE/DITTO invoked from CMS (as described in “CMS/VSE Mode: Starting VSE/DITTO from CMS” on page 14).

For more information about the FL function, use the online help.

FMT (Set DBCS Format)

Purpose Specify the format of records that contain double-byte character set (DBCS) data. The format that you specify affects all print commands.

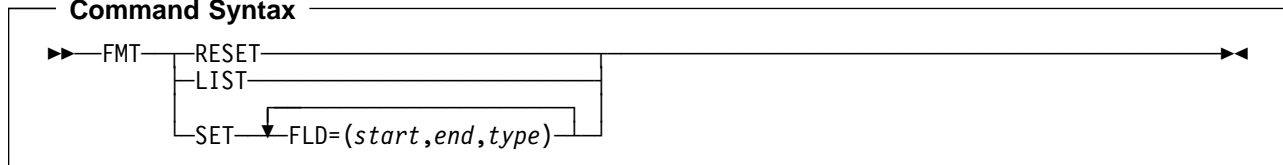
If you use the FMT function from the VSE or MVS system console, within a REXX/VSE procedure, or within a batch job, your format definition affects only the current session or batch job. In any other case, your format definition is saved in the user profile and is available for further DITTO sessions.

The FMT function and the DBCSPRT parameter of the SET function work together as follows:

- The FMT function specifies which data within each record is DBCS.
- The DBCSPRT parameter of the SET function specifies how DBCS data is printed.

For more information about the SET function, see “SET (Set Processing Options)” on page 242.

Command Syntax



Batch Syntax

\$\$DITTO FMT ...

Keyword Syntax

DITTO \$FMT ...

| MVS keywords | VSE keywords | CMS keywords |
|--|--|--|
| { RESET LIST SET,FLD=(start,end,type)[,...]} | { RESET LIST SET,FLD=(start,end,type)[,...]} | { RESET LIST SET,FLD=(start,end,type)[,...]} |

Parameters

LIST Lists all the field definitions, whether you defined them explicitly or DITTO used the default.

RESET Resets all field definitions to the default FLD=(1,*,EB).

SET Defines a new set of fields.

start Starting position of a field, in bytes, where 1 represents the start of the record.

end Ending position of the field, in bytes. Specify an asterisk (*) for the end of the record.

type Type of a field:

DB DBCS data. The field length must be an even number of characters.

MI Mixed DBCS and EBCDIC data. Data between the shift out character X'0E' and the shift in character X'0F' is treated as DBCS data. DITTO ignores any X'0F' (SI) code it finds at an even offset from the start of a DBCS subfield.

EB EBCDIC data (the default).

The field definitions must be in ascending order, and must not overlap. Any areas that you do not define are considered EBCDIC data.

You can define up to 32 fields. In line mode, you are prompted for up to 32 field definitions. To end prompting, enter a null response or specify an end position of *.

For example, the line-mode command:

```
FMT,SET,FLD=(10,27,MI),FLD=(40,59,DB)
```

gives you these field definitions:

```
FIELD1 (1,9,EB)
FIELD2 (10,27,MI)
FIELD3 (28,39,EB)
FIELD4 (40,59,DB)
FIELD5 (60,*,EB)
```

DITTO prints a mask of the record, with a scale to help you to adjust your definition:

```
EEEEEEEEEMMMMMMMMMMMMMMMMMMMMMEEEEEEEEEEEEEDBDBDBDBDBDBDBDBDBDBE*
1...5...10...5...20...5...30...5...40...5...50...5...60.
```

The valid double-byte hexadecimal values are X'0000', X'4040', and X'4141' to X'FEFE'. DITTO replaces all other pairs of double-byte characters with X'4040'.

Batch Example (MVS)

```
//FMT JOB (acct),'name' Define DBCS Record Fmt
//DITTO EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
$$DITTO FMT SET,FLD=(10,27,MI),FLD=(40,59,DB)
$$DITTO SET DBCS=SOSI
$$DITTO EOJ
//
```

Batch Example (VSE)

```
// JOB FMT - Define DBCS Record Format
// UPSI 1
// EXEC DITTO
$$DITTO FMT SET,FLD=(10,27,MI),FLD=(40,59,DB)
$$DITTO SET DBCS=3200
$$DITTO EOJ
/ &
```

FP (CMS File Print)—CMS Only

Purpose Print a CMS file in character or hexadecimal format.

Command Syntax

```
▶—FP—infile—format—skip—nlrecs—recsize—◀◀
```

Batch Syntax

```
$$DITTO FP ...
```

Keyword Syntax

```
DITTO $FP ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--------------|--|
| | | INFILE =' <i>infile</i> ' [FORMAT={CHAR HEX}] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [RECSIZE=recsize] |

Parameters

- format* The format of the output:
CHAR Character format (the default)
HEX Hexadecimal format.
- infile* Specifies the CMS input file identifier: *filename filetype [filemode]*, enclosed in single quotes. If the filemode is omitted, the normal minidisk search order is used.
- nlrecs* Number of records to be printed or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are printed.
- recsize* Length of the input records; this number is used for deblocking. The maximum is 9 999 999.
- skip* Number of records to be skipped from the beginning of the file. The default is 0.

Keyword Example (CMS)

```
/* REXX - Print CMS file */
"DITTO $FP INFILE 'PROFILE EXEC A',
"FORMAT HEX NLRECS 5"
Exit rc
```

FS (CMS File to Sequential Data)—CMS/VSE Mode Only

Purpose Copy records from a CMS file to a sequential file. This function is available only in full-screen mode VSE/DITTO invoked from CMS (as described in “CMS/VSE Mode: Starting VSE/DITTO from CMS” on page 14).

For more information about the FS function, use the online help.

FSF (Forward Space File)

Purpose Move the tape forward one or more tape files. This function moves the tape forward and stops immediately after reading the appropriate number of tape marks. Thus, the tape is positioned *after* the tape mark.

Command Syntax

```
▶▶—FSF—intape ┌───┐
                  │   │
                  └───┘ nfiles ▶▶
```

Batch Syntax

```
$$DITTO FSF ...
```

Keyword Syntax

```
DITTO $FSF ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--|--|---|
| [LABEL=BLP] {INPUT= <i>ddname</i> OUTPUT= <i>ddname</i> } [NFILES={1 <i>nfiles</i> }] | {INPUT={ <i>unit</i> SYS <i>nnn</i> } OUTPUT={ <i>unit</i> SYS <i>nnn</i> } } [NFILES={1 <i>nfiles</i> }] | {INPUT= <i>unit</i> OUTPUT= <i>unit</i> } [NFILES={1 <i>nfiles</i> }] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
| the first DITTO function that uses the tape. For BLP processing requirements, see
| "Customizing the Security Environment" in the *DITTO/ESA Installation and Customization*
| *Guide*.

nfiles Number of tape files to be spaced forward. The maximum is 99 999; the default is 1.
Leading tape marks are also counted. For more information on counting tape files, see
"Positioning a Tape" on page 68.

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or VOL1) of the tape volume to be mounted. For a volume set,
specify up to five volsers in the form (volser1, volser2...).

FSN

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form *[yy]yy.ddd*.

FSN (Full-Screen No)

Purpose Switch from full-screen mode to line mode, or start a new line mode DITTO session. Enter the command FSN from the command line in full-screen DITTO, or enter the command DITTO FSN from outside DITTO. This function is not available in batch mode.

FSR (Forward Space Record)

Purpose Move the tape forward one or more physical records (blocks).

Command Syntax

```
▶—FSR—intape [ nblks ] ▶▶
```

Batch Syntax

```
$$DITTO FSR ...
```

Keyword Syntax

```
DITTO $FSR ...
```

| MVS keywords | VSE keywords | CMS keywords |
|---|--|--|
| [LABEL=BLP] {INPUT= <i>ddname</i> OUTPUT= <i>ddname</i> } [NBLKS={ <u>1</u> <i>nblks</i> }] | {INPUT={ <i>unit</i> SYS <i>nnn</i> } OUTPUT={ <i>unit</i> SYS <i>nnn</i> }} [NBLKS={ <u>1</u> <i>nblks</i> }] | {INPUT= <i>unit</i> OUTPUT= <i>unit</i> } [NBLKS={ <u>1</u> <i>nblks</i> }] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
| the first DITTO function that uses the tape. For BLP processing requirements, see
| "Customizing the Security Environment" in the *DITTO/ESA Installation and Customization*
| *Guide*.

nblks Number of tape blocks to be spaced forward. The maximum is 99999999; the default is 1. A tape mark is considered a block.

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or V0L1) of the tape volume to be mounted. For a volume set, specify up to five volsers in the form (volser1, volser2...).

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form [*yy*]*yy.ddd*.

FSY (Full-Screen Yes)

Purpose Switch from line mode to full-screen mode. This function is available only in line mode. This function is ignored if full-screen mode is not available. Enter the command FSY in response to the DITTO function? prompt.

Command Syntax

▶—FSY—◀

Parameters: None.

FT (CMS File to Tape)—CMS Only

Purpose Copy records from a CMS file to a tape file.

Command Syntax

▶—FT—*infile*—*outtape*—*tapefile*—*skip*—*nlrecs*—*recfmout*—*blksize*—◀

Batch Syntax

\$\$DITTO FT ...

Keyword Syntax

DITTO \$FT ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--------------|---|
| | | INFILE='infile' OUTPUT=unit [DENS=mm] [OUTFILE=tapefile] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] |

Parameters

- blksize** If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.
- infile** Specifies the CMS input file identifier: *filename filetype [filemode]*, enclosed in single quotes. If the filemode is omitted, the normal minidisk search order is used.
- mm** A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.
- nlrecs** Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.
- outtape** Specification of the output tape unit. (Command Syntax only.) Specify: *unit[-mm]*.

recfmout Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

- U** Undefined length
- F** Fixed length
- V** Variable length
- D** Variable-length ISO/ANSI tape records
- B** Blocked
- S** Spanned format

skip Number of records to be skipped from the beginning of the file. The default is 0.

tapefile File ID of a tape file, if you want to perform label processing. By default, NL processing is performed.

unit Device number (3 or 4 hexadecimal digits).

Keyword Example (CMS)

```
/* REXX - copy CMS file to SL tape file */
"DITTO $REW OUTPUT 182" /* goto begin of tape */
"DITTO $FT INFILE 'PROFILE EXEC A' ,
"OUTPUT 182 OUTFILE PROFILE.EXEC"
Exit rc
```

FV (CMS File to VSAM)—CMS and CMS/VSE Mode Only

Purpose Copy records from a CMS file to a VSAM data set.

You can use this function:

- Under CMS, if you have CMS/VSAM.
- In CMS/VSE mode, to copy data from CMS to a VSE system.

Command Syntax (CMS only)

```
▶—FV—infile—outds—┐reuse┘┐skip┘┐nlrecl┘—▶
```

Batch Syntax (CMS only)

```
$$DITTO FV ...
```

Keyword Syntax (CMS only)

```
DITTO $FV ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--------------|---|
| | | INFILE='infile' [POSITION={0 skip}] [NLRECS={ALL nlrecl}] DSNOUT=dsname [UCATOUT=ucatl] [REUSE={NO YES}] [PWDOUT=pwdout] |

Parameters

| | |
|---------------|---|
| <i>dsname</i> | Name of a VSAM data set. |
| <i>infile</i> | Specifies the CMS input file identifier: <i>filename filetype [filemode]</i> , enclosed in single quotes. If the filemode is omitted, the normal minidisk search order is used. |
| <i>nlrecl</i> | Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>outds</i> | Specification of the VSAM output data set. (Command Syntax only.) Specify: <i>dsname,[ucatl]</i> . |
| <i>pwdout</i> | Password of a password-protected VSAM data set. |
| <i>reuse</i> | Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO. |
| <i>skip</i> | Number of records to be skipped from the beginning of the file. The default is 0. |
| <i>ucatl</i> | The data set name of the user catalog. |

II (Interpret Cards)

Purpose Interpret a prepunched card deck. The input/output device must be a card punch with the Read/Interpret feature installed.

- MVS: Input is SYSPUNCH. Output is DITIIPRT, which must be allocated with unit affinity to SYSPUNCH. An example of how you might allocate DITIIPRT with unit affinity to SYSPUNCH is shown below:

```
//DITIIPRT DD UNIT=AFF=SYSPUNCH
```

- VSE: Cards are read from and written to SYSPCH.
- CMS: Card are read from and written to the virtual punch address 00D.

| | |
|-----------------------|-----------------------|
| Command Syntax | |
| ▶—II—◀ | |
| Batch Syntax | Keyword Syntax |
| \$\$DITTO II | DITTO \$II |

Parameters: None.

INT (Initialize Tape)

Purpose Initialize tapes in IBM standard label format, ASCII label format, or NL tape.

| | |
|--|-----------------------|
| Command Syntax | |
| ▶—INT— <i>outtape</i> — <div style="display: flex; justify-content: space-around; margin-top: 5px;"> <i>volser</i> <i>tapefile</i> <i>ascii</i> <i>scale</i> </div> | |
| Batch Syntax | Keyword Syntax |
| \$\$DITTO INT ... | DITTO \$INT ... |

| MVS keywords | VSE keywords | CMS keywords |
|--|--|---|
| [LABEL=BLP] OUTPUT= <i>ddname</i> [DENS= <i>mm</i>] [VOLSER= <i>volser</i>] [OUTFILE= <i>tapefile</i>] [ASCII={YES NO}]] [SCALE= <i>scale</i>] | OUTPUT={ <i>unit</i> SYS <i>nnn</i> } [DENS= <i>mm</i>] [VOLSER= <i>volser</i>] [OUTFILE= <i>tapefile</i>] [ASCII={YES NO}]] [SCALE= <i>scale</i>] | OUTPUT= <i>unit</i> [DENS= <i>mm</i>] [VOLSER= <i>volser</i>] [OUTFILE= <i>tapefile</i>] [ASCII={YES NO}]] [SCALE= <i>scale</i>] |

Parameters

- ascii* Specify YES to format the tape with American National Standard labels for use with ASCII files. Specify NO to format the tape with EBCDIC labels. The default depends on the value of the SET TAPELBL parameter.
- ddname* Refers to a DD or TSO ALLOC statement.

INT

| | |
|-----------------|---|
| LABEL=BLP | Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the <i>DITTO/ESA Installation and Customization Guide</i> . |
| <i>mm</i> | A 2-byte tape mode code, as shown in "Tape Density and Mode Values" on page 67. |
| <i>outtape</i> | Specification of the output tape unit. (Command Syntax only.) |
| | MVS: |
| | <ul style="list-style-type: none">• If the tape is already allocated specify: <i>ddname[-mm]</i>.• If label processing is desired specify: <i>ddname[-mm],[tapevol],unit,{SL AL},tapedsn,[seqnum],[xdate]</i>.• For NL or BLP tape processing specify: <i>ddname[-mm],[tapevol],unit,{NL BLP},[seqnum],[xdate]</i>. |
| | VSE or CMS: |
| | <ul style="list-style-type: none">• Specify: <i>unit[-mm]</i>. |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapefile</i> | Name to be used to write the HDR1 label. The default is to create a HDR1 label with a dummy name for labeled tapes. This parameter is ignored for NL tapes. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted, or blank for a non-specific volume request. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number to be used for initialization. If omitted, an NL tape with no volser is produced. |
| <i>xdate</i> | Expiration date in the form <i>[yy]yy.ddd</i> . |
| <i>scale</i> | Limits the amount of data that can be stored on the tape to a percentage of its capacity. Allowed values are 0 (100% capacity; no limit applied), 25, 50, or 75. For example, SCALE=25 limits the tape to 25% percent of its normal capacity. Only valid for 3590 devices; it is not valid for a 3590 running in 3489E emulation mode. |

Batch Example (MVS)

```

//INT JOB (acct),'name' Initialize SL Tape
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DITTO EXEC DTOBAT
//SYSPRINT DD SYSOUT=*
//TAPE DD UNIT=561,VOL=SER=DIT001,
// DISP=SHR,LABEL=(,BLP)
//SYSIN DD *
$$DITTO INT OUTPUT=TAPE,VOLSER=DIT001,
$$DITTO SCALE=25,OUTFILE=MY.TEST.TAPE
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB INT - Initialize SL ASCII Tape
// UPSI 1
// EXEC DITTO
$$DITTO INT OUTPUT=580,VOLSER=DIT001,
$$DITTO OUTFILE='MY.TEST.TAPE',
$$DITTO ASCII=NO
$$DITTO EOJ
/*
/&

```

Keyword Example (CMS)

```

/* REXX - Initialize SL Tape */
"DITTO $INT OUTPUT 182 VOLSER 123456" ,
"OUTFILE MY.TEST.TAPE"

Exit rc

```

LB (Library Member Browse)—VSE Only

Purpose Browse a library member at the terminal. This function is available only in full-screen mode under VSE.

For more information about the LB function, use the online help or see "Browsing Data" on page 32.

LC (Library Member to Card)—VSE Only

Purpose Copy a library member to cards. Card output is written to SYSPCH.

If the library member is a phase, it can have any length. For any other type of library member, the record length (or string length) must be a multiple of 80.

Command Syntax

```
▶—LC—inlib—member—└┬—interpret—┘▶
```

Batch Syntax

```
$$DITTO LC ...
```

Keyword Syntax

```
DITTO $LC ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | LIBIN= <i>lib</i> MEMBERIN= <i>member</i> [INTERPRET={NO YES}] | |

Parameters

inlib Specification of the input library. (Command Syntax only.)

- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
- For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
- For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucatl]*.

interpret Interpret the output. If you specify YES, the output device must be a card punch with the card print feature installed. The default is NO.

lib Library name, followed by a period and the sublibrary name (*library.sublibrary*).

libdsn The data set name of the library. If it contains blanks or commas, enclose it in quotes.

member Name of the library member in the form *name.type*.

ucatl The data set name of the user catalog.

volser Volume serial number of the device where the library is located.

Batch Example (VSE)

```
// JOB LC - Library Member to Card
// UPSI 1
// EXEC DITTO
$$DITTO LC LIBIN=PRD1.MACLIB,MEMBERIN=DTFCD.A
$$DITTO EOJ
/&
```

LDEL (Library Member Delete)—VSE Only

Purpose Delete a library member or group of library members.

Command Syntax

```
▶—LDEL—inlib—memmask—◀◀
```

Batch Syntax

```
$$DITTO LDEL ...
```

Keyword Syntax

```
DITTO $LDEL ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | LIBIN = <i>lib</i> MEMBERIN = <i>memmask</i> | |

Parameters

- inlib* Specification of the input library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucat]*.
- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- memmask* A generic member name in the form *name.type*. Within *name* or *type*, you can include a percent sign (%) to represent exactly one character, or an asterisk (*) to represent any number of characters (or no characters).
- ucat* The data set name of the user catalog.
- volser* Volume serial number of the device where the library is located.

Batch Example (VSE)

```
// JOB LDEL - Delete all members with type OLDA
// UPSI 1
// EXEC DITTO
$$DITTO LDEL LIBIN=PRD2.SAVE,MEMBERIN=*.OLDA
$$DITTO EOJ
/&
```

LDL (Library Directory List)—VSE Only

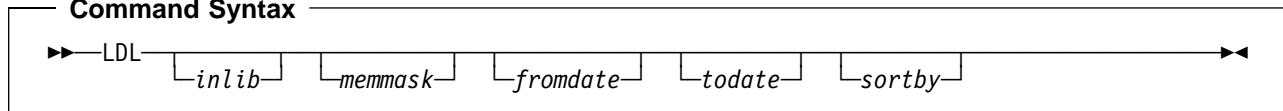
Purpose Work with a list of library members.

In full-screen mode, you can print or display the list; if you display the list, you can browse, print, update, edit or delete any of the listed members. In any other mode, you can only print the list.

The information can be sorted by member name, library name (then member name within each library), member type, member size, or last update date.

For more information about using this function in full-screen mode, see “Using the LDL Function in Full-Screen Mode” on page 125.

Command Syntax



Batch Syntax

\$\$DITTO LDL ...

Keyword Syntax

DITTO \$LDL ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | [LIBIN= <i>libmask</i>] [...] [MEMBERIN={*.}* <i>memmask</i>] [FROMDATE= <i>fromdate</i>] [TODATE= <i>todate</i>] [SORTBY={ <u>NAME</u> LIB TYPE DATE SIZE}] | |

Parameters

fromdate The earliest update date, in the form *yyyymmdd*. By default, all members are listed regardless of their update date.

inlib Specification of the input library. (Command Syntax only.) You are prompted for library names until you enter a null string, or your eighth library name.

- If a DLBL statement or standard labels are available, specify: *libmask*.
- For a library in BAM-managed space, specify: *libmask,libdsn,volser*.
- For a library in VSAM-managed space, specify: *libmask,libdsn,,[ucaf]*.

If you omit the library specification, all libraries stored in the system standard labels are scanned.

libdsn The data set name of the library. If it contains blanks or commas, enclose it in quotes.

libmask Library name, optionally followed by a period and the sublibrary name (*library.sublibrary*). Within the sublibrary name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters). By default, all the sublibraries within the library are used.

You can supply up to eight LIBIN keywords. If you omit the LIBIN keyword, all libraries stored in the system standard labels are scanned.

memmask A generic member name in the form *name.type*. Within *name* or *type*, you can include a percent sign (%) to represent exactly one character, or an asterisk (*) to represent any number of characters (or no characters). The default is *.*

sortby One of the following:

NAME Sorts by member name (the default)

LIB Sorts by library name, then member name within each library
TYPE Sorts by member type
DATE Sorts by last update date
SIZE Sorts by member size.

todate The latest update date, in the form *yyyymmdd*.

ucat The data set name of the user catalog.

volser Volume serial number of the device where the library is located.

Batch Example (VSE)

```
// JOB LDL - List all phases in PRD1.BASE
// UPSI 1
// EXEC DITTO
$$DITTO LDL LIBIN=PRD1.BASE,MEMBERIN=*.PHASE,
$$DITTO SORTBY=DATE
$$DITTO EOJ
/&
```

LE (Library Member Edit)—VSE Only

Purpose Edit a library member at the terminal. This function is only available in full-screen mode. For members with string record format, the LU function is invoked.

For more information about the LE function, use the online help, or see Chapter 4, “Editing Data in Full-Screen Mode” on page 39.

LF (Library Member to CMS File)—CMS/VSE Mode Only

Purpose Copy records from a library member to a CMS file. This function is available only in full-screen mode VSE/DITTO invoked from CMS (as described in “CMS/VSE Mode: Starting VSE/DITTO from CMS” on page 14).

For more information about the LF function, use the online help.

LL (Library Member to Library Member)—VSE Only

Purpose Copy or move a member or group of members to another library or sublibrary, or to different names in the same library and sublibrary.

Command Syntax

```
▶▶ LL inlib memmask1 move outlib memmask2 replace ▶▶
```

Batch Syntax

```
$$DITTO LL ...
```

Keyword Syntax

```
DITTO $LL ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | LIBIN=lib1 MEMBERIN=memmask1 [MOVE={YES NO}] LIBOUT=lib2 [MEMBEROUT=memmask2] [REPLACE={YES NO}] | |

Parameters

- inlib** Specification of the input library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,[ucaf]*.
- libdsn** The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- lib1** Source library name, followed by a period and the sublibrary name (*library.sublibrary*).
- lib2** Target library name, followed by a period and the sublibrary name (*library.sublibrary*).
- memmask1** A generic source member name in the form *name.type*. Within *name* or *type*, you can include a percent sign (%) to represent exactly one character, or an asterisk (*) to represent any number of characters (or no characters).
- memmask2** A generic target member name in the form *name.type*. Within *name* or *type*, you can include a percent sign (%) to represent one character from the equivalent position in the source member name, or an asterisk (*) to represent the rest of the source member name (starting from the equivalent position). If you omit *memmask2*, the members are copied with unchanged names.
- For example, if MEMBERIN=DITMOD.PHASE and MEMBEROUT=OLD%%%.*SAV, a member is created with the name OLDMOD.PHASESAV.
- move** Specifies whether input members are to be moved or copied to the output library.
- YES** Deletes the member from the input library after copying it to the output library, except in the following case: if the member already exists in the output library, and you specify REPLACE=NO, then no move occurs (the input and output library members are left unchanged).
- NO** Leaves the member in the input library after copying it to the output library (the default).

| | |
|----------------|--|
| <i>outlib</i> | Specification of the output library. (Command Syntax only.) <ul style="list-style-type: none"> • If a DLBL statement or standard labels are available, specify: <i>library.sublibrary</i>. • For a library in BAM-managed space, specify: <i>library.sublibrary,libdsn,volser</i>. • For a library in VSAM-managed space, specify: <i>library.sublibrary,libdsn,,[ucatl]</i>. |
| <i>replace</i> | Specifies whether or not to replace an existing member of the same name in the output library. |
| | YES Replace the member in the output library. |
| | NO Do not replace the member in the output library. If you specify MOVE=YES and REPLACE=NO, and the member already exists in the output library, then no move occurs (the input and output library members are left unchanged). |
| | |
| <i>ucatl</i> | The data set name of the user catalog. |
| <i>volser</i> | Volume serial number of the device where the library is located. |

Batch Example (VSE)

```

// JOB LL - Move all new members to LIB2
// UPSI 1
// EXEC DITTO
// DLBL USERCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL DITTEST,'DITTEST.WORK.LIB',,VSAM,
      CAT=USERCAT
$$DITTO VER
$$DITTO LL LIBIN=DITTEST.LIB1,
$$DITTO MEMBERIN=DTF*.SAVEA,MOVE=YES,
$$DITTO LIBOUT=DITTEST.LIB2,
$$DITTO MEMBEROUT=*.SAVEB,REPLACE=NO
$$DITTO EOJ
/*
/&
* $$ EOJ

```

LOAD (Load CMS Files from Tape)—CMS Only

Purpose Load files from one of the following:

- A DITTO dump tape (created with the DUMP command in DITTO)
- A minidisk from a DASD Dump Restore (DDR) tape or file
- A minidisk from a fullpack DDR tape or file (or from a multivolume backup tape generated by one DDR run with continuation tapes or several consecutive DDR runs).

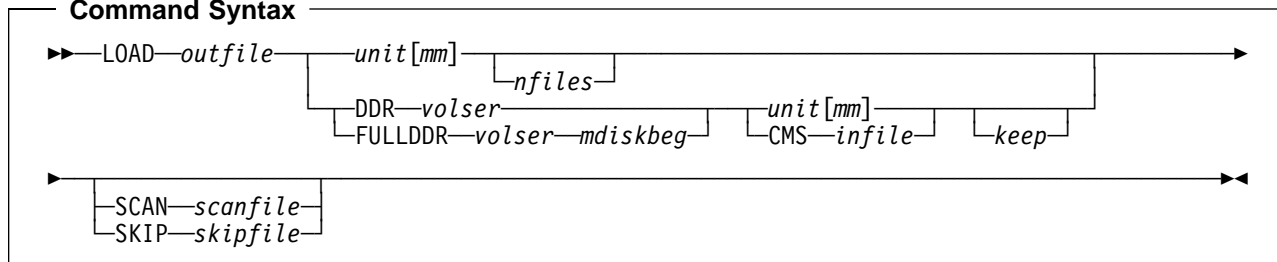
When you load from a DDR or fullpack DDR tape or file, the minidisk that you specify is first loaded into a staging file. The files that you specify are then copied from the staging file to disk. You can optionally keep this staging file and use it for subsequent load operations.

You can load files from a minidisk that originally existed on a different type of disk. For example, you can dump an FBA disk to tape with DDR and later use the DITTO LOAD command to restore a minidisk from the DDR tape to a CKD disk.

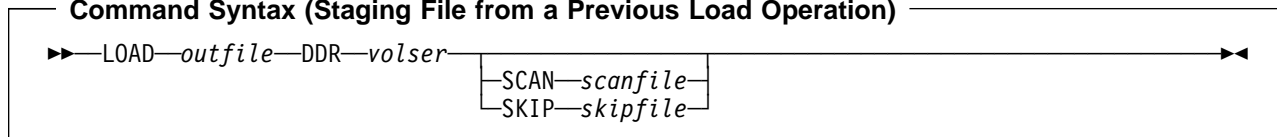
In command mode, the following return codes are set:

- 0 Successful completion
- 4 File(s) not found
- 12 Target or loadfile disk not R/W
- 13 Target or loadfile disk is full
- 16 Severe error (indicated by messages).

Command Syntax



Command Syntax (Staging File from a Previous Load Operation)



Batch Syntax

\$\$DITTO LOAD ...

Keyword Syntax

DITTO \$LOAD ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--------------|--|
| | | <pre> OUTFILE='outfile' [TYPE={DUMP DDR FULLDDR}] [{ INPUT=unit INPUT=CMS INFILE='infile' }] [NFILES={1 nfiles}] [VOLSER=volser] [BEGIN=mdiskbeg] [KEEP={NO YES}] [{ SCAN='scanfile' SKIP='skipfile' }] </pre> |

Parameters

- CMS** Specify CMS if the DDR input is loaded from a CMS file.
- infile** Specifies the CMS input file identifier: *filename filetype [filemode]*, enclosed in single quotes. If the filemode is omitted, the normal minidisk search order is used.
- keep** Specify YES to keep the staging file as input for subsequent loading operations. By default, the staging file is erased at the end of the LOAD operation.
- mdiskbeg** The beginning disk address (start cylinder or FBA block number) of the minidisk that you want to load from a fullpack DDR tape or file.
- mm** A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.
- nfiles** Number of tape files to be loaded. The maximum is 99 999; the default is 1.
- outfile** File identifier of the file or files that you want to load: *filename filetype [filemode]*, enclosed in single quotes. Within the filename or filetype, you can include a percent sign (%) to represent exactly one character, or an asterisk (*) to represent any number of characters (or no characters). The default filemode is A.
- scanfile** Name and type of a file. DITTO skips all preceding files, and starts loading with the specified file.
If you specify * *, DITTO lists the files on the tape. No files are loaded.
- skipfile** Name and type of a file. DITTO skips all files up to and including the specified file, and starts loading with the file *after* the specified file.
If you specify * *, DITTO lists the files on the tape. No files are loaded.
- type** The type of input tape: DUMP, DDR, or FULLDDR. The default is DUMP.
- unit** Device number (3 or 4 hexadecimal digits). Specify CMS if the DDR input is loaded from a CMS disk; omit if a staging file from a previous DDR load exists.
- volser** Volume identifier of the minidisk that you want to load. The entire minidisk is loaded to a staging file with the name *volser* DITLOAD A.
If you are loading files from a fullpack DDR tape or file, the *mdiskbeg* parameter (not the *volser* parameter) determines which minidisk is loaded. In this case, *volser* is used for verification only. To skip verification, omit the *VOLSER=volser* keyword or specify SCRATCH if you are not using keywords.
In line mode, if the *volser* that you specify matches the name of a staging file that you kept from a previous load operation, files are loaded from the staging file and you are not prompted for an input device or the *keep* parameter.

Keyword Example (CMS)

```
/* REXX - LOAD CMS File from Tape */
/* Load a DITTO dump tape */
"DITTO $LOAD OUTFILE '* * Z' TYPE DUMP" ,
    "INPUT 181"
Exit rc
```

Keyword Example (CMS)

```
/* REXX - LOAD CMS File from Tape */
/* Load a DDR tape and keep the staging file */
"DITTO $LOAD OUTFILE '* * Z' TYPE DDR" ,
    "VOLSER ZIE191 INPUT 181 KEEP YES"
Exit rc
```

Keyword Example (CMS)

```
/* REXX - LOAD CMS file from tape */
/* Load from existing staging file ... */
/* ... kept by a previous LOAD operation */
"DITTO $LOAD OUTFILE '* EXEC B' TYPE DDR" ,
      "VOLSER ZIE191"
Exit rc
```

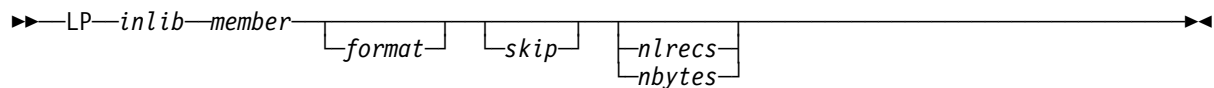
Keyword Example (CMS)

```
/* REXX - LOAD CMS file from tape */
/* Load a fullpack DDR tape */
"DITTO $LOAD OUTFILE '* * Z' TYPE FULLDDR" ,
      "VOLSER ZIE191 BEGIN 675" ,
      "INPUT 181"
Exit rc
```

LP (Library Member Print)—VSE Only

Purpose Print a library member in character or hexadecimal format.

Command Syntax



Batch Syntax

`$$DITTO LP ...`

Keyword Syntax

`DITTO $LP ...`

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | LIBIN = <i>lib</i> MEMBERIN = <i>member</i> [FORMAT={CHAR HEX}] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [NBYTES={ALL nbytes}] | |

Parameters

- format** The format of the output:
 - CHAR** Character format (the default)
 - HEX** Hexadecimal format.
- inlib** Specification of the input library. (Command Syntax only.)
 - If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[uca]*.
- lib** Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn** The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- member** Name of the library member in the form *name.type*.
- nbytes** For a library member in string format, the number of bytes to be printed or ALL. The maximum number is 99 999 999. If you specify ALL or you omit the parameter, all the remaining bytes are printed.
- nlrecs** For a library member in fixed format, the number of records to be printed or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are printed.
- skip** Number of bytes (if the library member is stored in string format) or number or records (otherwise) to be skipped from the beginning of the member. The default is 0.

ucat The data set name of the user catalog.
volser Volume serial number of the device where the library is located.

Batch Example (VSE)

```
// JOB LP - Print a library member
// UPSI 1
// EXEC DITTO
$$DITTO LP LIBIN=PRD2.CONFIG,
$$DITTO MEMBERIN=ATCSTR00.B
$$DITTO EOJ
/&
```

LR (Library Member Rename)—VSE Only

Purpose Rename a library member or a group of library members.

Command Syntax

```
▶—LR—inlib—memmask1—memmask2—◀◀
```

Batch Syntax

```
$$DITTO LR ...
```

Keyword Syntax

```
DITTO $LR ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | LIBIN = <i>lib</i> MEMBERIN = <i>memmask1</i> NEWNAME = <i>memmask2</i> | |

Parameters

- inlib* Specification of the input library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucatl]*.
- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- memmask1* A generic old member name in the form *name.type*. Within *name* or *type*, you can include a percent sign (%) to represent exactly one character, or an asterisk (*) to represent any number of characters (or no characters).
- memmask2* A generic new member name in the form *name.type*. Within *name* or *type*, you can include a percent sign (%) to represent one character from the equivalent position in the old member name, or an asterisk (*) to represent the rest of the old member name (starting from the equivalent position).
- For example, if MEMBERIN=DITMOD.PHASE and MEMBEROUT=OLD%%%.*SAV, the member is renamed OLDMOD.PHASESAV.
- ucatl* The data set name of the user catalog.
- volser* Volume serial number of the device where the library is located.

Batch Example (VSE)

```
// JOB LR - Rename library members
// UPSI 1
// DLBL USERCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL DITTEST,'DITTEST.WORK.LIB',,VSAM,
    CAT=USERCAT
// EXEC DITTO
$$DITTO LR LIBIN=DITTEST.LIB1,
$$DITTO MEMBERIN=DTF*.SAVEA,
$$DITTO NEWNAME=*.OLD
$$DITTO EOJ
/*
/&
* $$ EOJ
```


| | |
|-----------------|--|
| <i>inlib</i> | Specification of the input library. (Command Syntax only.) <ul style="list-style-type: none"> • If a DLBL statement or standard labels are available, specify: <i>library.sublibrary</i>. • For a library in BAM-managed space, specify: <i>library.sublibrary,libdsn,volser</i>. • For a library in VSAM-managed space, specify: <i>library.sublibrary,libdsn,,[ucaf]</i>. |
| <i>lib</i> | Library name, followed by a period and the sublibrary name (<i>library.sublibrary</i>). |
| <i>libdsn</i> | The data set name of the library. If it contains blanks or commas, enclose it in quotes. |
| <i>member</i> | Name of the library member in the form <i>name.type</i> . |
| <i>outds</i> | Specification of the sequential output data set. (Command Syntax only.) <ul style="list-style-type: none"> • If a DLBL statement for SDSKOUT is provided, no input is required. • For existing VSAM-managed SAM specify: <i>dsname,YES,[ucaf],OLD</i>. • For new VSAM-managed SAM specify: <i>dsname,YES,[ucaf],NEW,space,avgsz</i>. • For BAM-managed SAM specify: <i>dsname,NO,{unit Vvolser},begin,end</i>. |
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <p>U Undefined length F Fixed length V Variable length B Blocked</p> |
| <i>space</i> | The number of space units to be allocated for a new data set. Specify <i>n</i> or (<i>n,m</i>), where <i>n</i> is the number of primary units, and <i>m</i> is the number of secondary units. The space unit is the length of an average record. |
| <i>SYSnnn</i> | Programmer logical unit to which the output device is assigned. |
| <i>ucat</i> | For VSAM-managed SAM or library space, the data set name of the user catalog. |
| <i>unit</i> | Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number of the device where the library is located. |
| <i>vsamout</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the output volume. |

LSL (Library Status List)—VSE Only

Purpose Display status information about a library, and list its sublibraries.

In full-screen mode, you can print or display the information; if you display the information, you can display the disk extents for the library or invoke the LDL function for any sublibrary. In any other mode, you can only print the list.

For more information about using this function in full-screen mode, see “Using the LSL Function in Full-Screen Mode” on page 127.

Command Syntax

```
▶▶—LSL—inlib—◀◀
```

Batch Syntax

```
$$DITTO LSL ...
```

Keyword Syntax

```
DITTO $LSL ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|-----------------------|--------------|
| | LIBIN= <i>libmask</i> | |

Parameters

- inlib* Specification of the input library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *libmask*.
 - For a library in BAM-managed space, specify: *libmask,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *libmask,libdsn,,[ucaf]*.
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- libmask* Library name, optionally followed by a period and the sublibrary name (*library.sublibrary*). Within the sublibrary name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters). By default, all the sublibraries within the library are shown.
- ucaf* The data set name of the user catalog.
- volser* Volume serial number of the device where the library is located.

Batch Example (VSE)

```
// JOB LSL - List information about library PRD1
// UPSI 1
// EXEC DITTO
$$DITTO LSL LIBIN=PRD1
$$DITTO EOJ
/&
```

LT (Library Member to Tape)—VSE Only

Purpose Copy a library member to tape.

Command Syntax

```
▶—LT—inlib—member—outtape—_tapefile_—_recfmout_—_blksize_—▶
```

Batch Syntax

```
$$DITTO LT ...
```

Keyword Syntax

```
DITTO $LT ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | LIBIN = <i>lib</i> MEMBERIN = <i>member</i> OUTPUT ={ <i>unit</i> SYS <i>nnn</i> } [DENS= <i>mm</i>] [{ FILEOUT={TAPEOUT <i>tlbl</i> OUTFILE= <i>tapefile</i> }] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE= <i>blksize</i>] | |

Parameters

blksize For a library member in string format, the size of the blocks on the tape. Any value that you specify for the record format is ignored. The default block size is the member size.

For a library member in fixed format, the actual block size (if *recfmout* contains F) or the maximum block size (otherwise). If *recfmout* contains B, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.

inlib Specification of the input library. (Command Syntax only.)

- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
- For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
- For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucaf]*.

lib Library name, followed by a period and the sublibrary name (*library.sublibrary*).

libdsn The data set name of the library. If it contains blanks or commas, enclose it in quotes.

member Name of the library member in the form *name.type*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

outtape Specification of the output tape unit. (Command Syntax only.) Specify: *unit*[**-***mm*].

recfmout For a library member in fixed format, the record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

- U** Undefined length
- F** Fixed length
- V** Variable length
- D** Variable-length ISO/ANSI tape records
- B** Blocked
- S** Spanned format

For a library member in string format, the output record format is always U.

| | |
|-----------------|--|
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapefile</i> | File ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed. |
| <i>tlbl</i> | Refers to a TLBL statement to process a labeled tape. The default is TAPEOUT. |
| <i>ucat</i> | The data set name of the user catalog. |
| <i>unit</i> | Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number of the device where the library is located. |

Batch Example (VSE)

```

// JOB LT - Library Member to SL Tape
// ASSGN SYS010,580
// TLBL MYTAPE,'TEST.OUTPUT',0,DIT001,,1
// UPSI 1
// EXEC DITTO
$$DITTO LT LIBIN=PRD1.BASE,
$$DITTO MEMBERIN=DITTO.PHASE,OUTPUT=SYS010,
$$DITTO FILEOUT=MYTAPE
$$DITTO EOJ
/*
/&

```

LU (Library Member Update)—VSE Only

Purpose Browse and update a library member at the terminal. This function is available only in full-screen mode under VSE.

For more information about the LU function, use the online help or see “Browsing and Updating Library Members in Full-Screen Mode” on page 129.

LV (Library Member to VSAM)—VSE Only

Purpose Copy a library member to a VSAM data set.

Command Syntax

```
▶▶ LV inlib member outds [reuse] [blksize] ▶▶
```

Batch Syntax

```
$$DITTO LV ...
```

Keyword Syntax

```
DITTO $LV ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | LIBIN= <i>lib</i> MEMBERIN= <i>member</i> [{ FILEOUT= {VDSKOUT <i>dlbl</i> } DSNOUT= <i>dsname</i> [UCATOUT= <i>ucat</i>] }] [REUSE= {NO YES}] [PWDOUT= <i>pwdout</i>] [BLKSIZE= <i>blksize</i>] | |

Parameters

- blksize* For a library member in string format, the size of the records in the VSAM data set. The default block size is the member size. For a library member in fixed format, this parameter is ignored and the records have a length of 80.
- dlbl* Refers to a DLBL statement. The default is VDSKOUT.
- dsname* Name of a VSAM data set.
- inlib* Specification of the input library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucat]*.
- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- member* Name of the library member in the form *name.type*.
- outds* Specification of the VSAM output data set. (Command Syntax only.)
- If a DLBL statement for VDSKOUT is provided, no input is required.
 - If no DLBL statement is provided specify: *dsname,[ucat]*.
- pwdout* Password of a password-protected VSAM data set.
- reuse* Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO.
- ucat* The data set name of the user catalog.
- volser* Volume serial number of the device where the library is located.

Batch Example (VSE)

```
// JOB LV - Library Member to VSAM
// DLBL TSTESDS, 'DITTO.TEST.ESDS',,VSAM
// UPSI 1
// EXEC DITTO
$$DITTO LV LIBIN=PRD2.CINFIG,
$$DITTO MEMBERIN=VTMAPPL.B,
$$DITTO FILEOUT=TSTESDS
$$DITTO EOJ
/&
```

LVL (Display Service Level)

Purpose Display the current level of DITTO you are using, together with the latest PTF applied. This function is the same as the VER (version) function.

Command Syntax

```
▶▶—LVL—◀◀
```

Batch Syntax

```
$$DITTO LVL
```

Keyword Syntax

```
DITTO $LVL
```

Parameters: None.

LX (Library Member to REXX Variable)—VSE only

Purpose Copy data from a library member into a REXX stem variable. Each record is copied to a variable named *stem.nnn*. (If the member is stored in string format, the entire member is considered one record.) *stem.0* is a counter.

For example, if you copy 3 records, *stem* is VARXX. and VARXX.0 has the value 5:

- The records are copied to VARXX.6, VARXX.7, and VARXX.8.
- The value of VARXX.0 is reset to 8.

This function is available only from a REXX procedure.

Command Syntax

```

▶▶ LX—inlib—member—stem—skip—nlrecs—nbytes

```

Keyword Syntax

DITTO \$LX ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | LIBIN= <i>lib</i> MEMBERIN= <i>member</i> VARNAME= <i>stem</i> [POSITION={0 <i>skip</i> }] [NLRECS={ALL <i>nlrecs</i> }] [NBYTES={ALL <i>nbytes</i> }] | |

Parameters

- inlib* Specification of the input library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucat]*.
- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- member* Name of the library member in the form *name.type*.
- nbytes* For a library member in string format, the number of bytes to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or you omit the parameter, all the remaining bytes are copied.
- nlrecs* For a library member in fixed format, the number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.
- skip* Number of bytes (if the library member is stored in string format) or number or records (otherwise) to be skipped from the beginning of the member. The default is 0.
- stem* A REXX stem variable name. The maximum length is 44 characters.
- ucat* The data set name of the user catalog.
- volser* Volume serial number of the device where the library is located.

Keyword Example (VSE)

```
/* REXX - copy library member to a rexx stem */  
"DITTO $LX LIBIN=PRD1.BASE," ,  
    "MEMBERIN=DITPROFL.A,VARNAME=RECS."  
Do i=1 To recs.0  
    Say recs.i /* show library member records */  
End  
Exit
```

MB (Memory Browse)

Purpose Browse user storage at the terminal. This function is available only in full-screen mode.

For more information about the MB function, use the online help or see “Browsing Data” on page 32.

OB (Object Browse)—MVS Only

Purpose Browse an OAM object at the terminal. This function is available only in full-screen mode under MVS.

For more information about the OB function, use the online help or see “Browsing Data” on page 32.

ODL (Object Directory List)—MVS Only

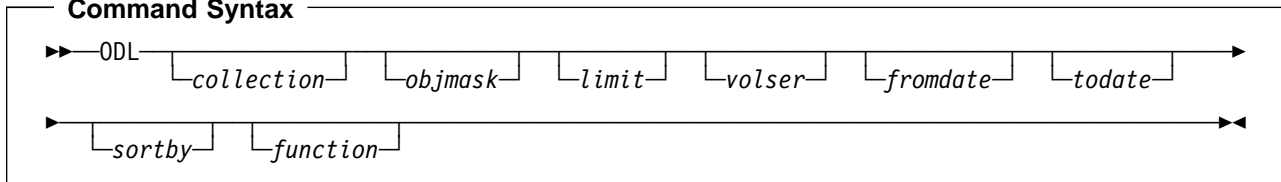
Purpose Work with a list of OAM objects from the specified collection or optical disk.

In full-screen mode, you can print or display the list; if you display the list, you can browse, print, update, or erase any of the listed objects. In command mode, you can invoke the full-screen mode ODL function. In any other mode, you can only print the list.

The information can be sorted by object name, size, or date of creation or expiration. You can also limit the objects that are listed, based on collection name, object name, location, or creation date.

For more information about using this function in full-screen mode, see “Using the ODL Function in Full-Screen Mode” on page 135.

Command Syntax



Batch Syntax

\$\$DITTO ODL ...

Keyword Syntax

DITTO \$ODL ...

| MVS keywords | VSE keywords | CMS keywords |
|---|--------------|--------------|
| [COLLECTN={** <i>collection</i> }] [OBJNAME={** <i>objmask</i> }] [LIMIT={DASD OPTICAL VOLUME}] [VOLSER= <i>volser</i>] [FROMDATE= <i>fromdate</i>] [TODATE= <i>todate</i>] [SORTBY={NAME DATE EDATE SIZE}] [FUNCTION={LIST PRINT}] | | |

Parameters

collection A collection name. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is all collections; in this case, you must specify LIMIT=VOLUME and a *volser*.

fromdate The earliest creation date, in the form *yyyymmdd*. By default, all objects are listed regardless of their creation date.

function The command that you want to execute:

- LIST** Display the list (the default)
- PRINT** Print the list.

limit To limit the list based on where objects are located, specify one of the following:

- DASD** Only objects that are stored on DASD
- OPTICAL** Only objects that are stored on optical disks
- VOLUME** Only objects that are stored on a specified optical disk. You must also specify the *volser* of the optical disk.

objmask An object name or generic object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is **. In batch mode, this means all objects; in any other mode, this means all objects that have the user's TSO prefix as high-level qualifier.

sortby One of the following:

NAME Sorts by object name (the default)

DATE Sorts by creation date

EDATE Sorts by expiration date

SIZE Sorts by object size.

today The latest creation date, in the form *yyyymmdd*.

volser If you specified VOLUME for *limit*, the volume serial number of the optical disk.

Batch Example (MVS)

```
//JOB (acct),'name',Optical Volume Contents
//S1 EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
$$DITTO ODL LIMIT=VOLUME,VOLSER=VBKUP6
$$DITTO EOJ
//
```

OE (Object Erase)—MVS Only

Purpose Erase an OAM object.

Command Syntax

```
▶—OE—collection—object—◀◀
```

Batch Syntax

```
$$DITTO OE ...
```

Keyword Syntax

```
DITTO $OE ...
```

| MVS keywords | VSE keywords | CMS keywords |
|---|--------------|--------------|
| COLLECTN= <i>collection</i> OBJNAME= <i>object</i> | | |

Parameters

collection Name of the OAM object collection. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

object Name of the OAM object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

Batch Example (MVS)

```
//S1      EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
$$DITTO OE COLLECTN=SYSU.OAM.CLLCT001,
$$DITTO   OBJNAME=DTT.ITT.WORK12
$$DITTO EOJ
/*
//
```


OO (Object to Object)—MVS Only

Purpose Copy an OAM object to another object collection, or to a different name in the same collection.

Command Syntax

```
▶▶ OO collection1 object1 [collection2] [object2] [storclas] [mgmtclas] ▶▶
```

Batch Syntax

```
$$DITTO OO ...
```

Keyword Syntax

```
DITTO $OO ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--|--------------|--------------|
| COLLECTN= <i>collection1</i> OBJNAME= <i>object1</i> [COLLECTN= <i>collection2</i>] [OBJNAME= <i>object2</i>] [STORCLAS= <i>storclas</i>] [MGMTCLAS= <i>mgmtclas</i>] | | |

Parameters

- collection1* Name of the OAM object collection that contains the object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
- collection2* Name of the OAM object collection to which you are copying the object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
- By default, the collection name is not changed. (In this case, you must change the object name.)
- mgmtclas* Name of an SMS management class.
- object1* Current name of the OAM object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
- object2* New name of the OAM object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
- By default, the object name is not changed. (In this case, you must change the collection name.)
- storclas* Name of an SMS storage class.

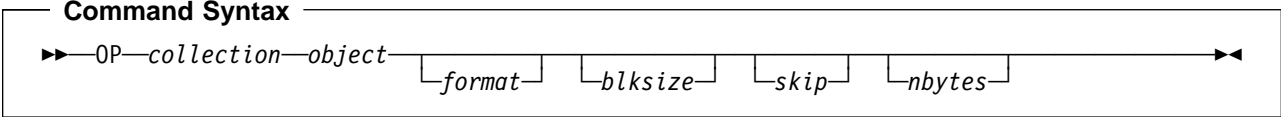
Note: The same keywords are used for input and output. The first occurrences of COLLECTN and OBJNAME refer to the input object; the second occurrences refer to the output object.

Keyword Example (MVS)

```
/* REXX */
/* Copy object within a collection ... */
"DITTO $00 COLLECTN='SYSU.OAM.CLLCT000',"
      "OBJNAME='DTT.ITT.WORK12',"
      "OBJNAME='DTT.ITT.COPY12'"
/* Copy object to other collection... */
"DITTO $00 COLLECTN='SYSU.OAM.CLLCT000',"
      "OBJNAME='DTT.ITT.COPY12',"
      "COLLECTN='SYSU.OAM.CLLCT001'"
exit
```

OP (Object Print)—MVS Only

Purpose Print an OAM object in hexadecimal or character format.



Batch Syntax

```

    $$DITTO OP ...
  
```

Keyword Syntax

```

    DITTO $OP ...
  
```

| MVS keywords | VSE keywords | CMS keywords |
|--|--------------|--------------|
| COLLECTN= <i>collection</i> OBJNAME= <i>object</i> [FORMAT={CHAR HEX}] [BLKSIZE={4096 <i>blksize</i> }] [POSITION={0 <i>skip</i> }] [BYTES={ALL <i>nbytes</i> }] | | |

Parameters

- blksize* Logical record size to be used to deblock the OAM object into logical records. The maximum is 32 768; the default is 4 096.
- collection* Name of the OAM object collection. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
- format* The format of the output:
CHAR Character format (the default)
HEX Hexadecimal format.
- nbytes* Maximum number of bytes to be printed or ALL. If this number is exceeded, printing stops. This parameter provides protection against printing an OAM object that is much larger than expected. The maximum number is 99 999 999. If you specify ALL or you omit the parameter, the entire object is printed.
- object* Name of the OAM object. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
- skip* Number of bytes to be skipped from the beginning of the object. The maximum is 99 999 999; the default is 0.

Batch Example (MVS)

```

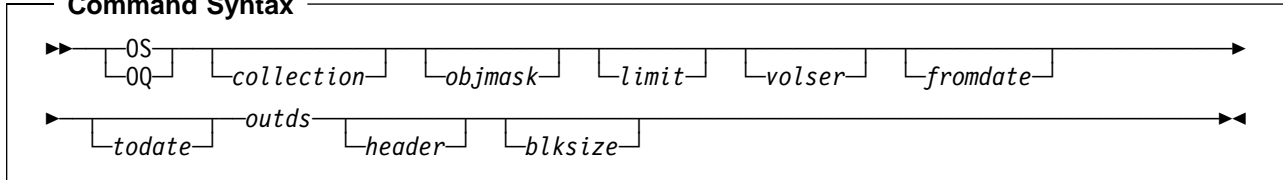
//JOB (acct),'name',Print OAM Object
//S1 EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
$$DITTO OP COLLECTN='SYSU.OAM.CLLCT000',
$$DITTO OBJNAME='DTT.ITT.TBL21',
$$DITTO BYTES=1000
$$DITTO EOJ
//
  
```

OS, OQ (Object to Sequential Data)—MVS Only

Purpose Copy one or more OAM objects to a sequential data set. For more information about using this function, see “Backing up OAM Objects in a Sequential or VSAM Data Set” on page 138.

For compatibility with previous releases, you can specify either OS or OQ.

Command Syntax



Batch Syntax

```
$$DITTO OS|OQ ...
```

Keyword Syntax

```
DITTO $OS|$OQ ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--|--------------|--------------|
| <pre> [COLLECTN={** collection}] [OBJNAME={** objmask}] [LIMIT={DASD OPTICAL VOLUME}] [VOLSER=volser] [FROMDATE=fromdate] [TODATE=todate] [{ OUTPUT={QSAMOUT ddname} DSNOUT=dsname [(member)] [VOLSEROUT=volser] [UNITOUT=unit] [DISP={NEW OLD MOD CAT}] [SPACE=space }] [HEADER={YES NO}] [BLKSIZE={4096 blksize}] </pre> | | |

Parameters

- blksize** Block size for the output data set. The maximum is 32760; the default is 4096.
- collection** A collection name. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
The default is all collections; in this case, you must specify LIMIT=VOLUME and a *volser*.
- ddname** Refers to a DD or TSO ALLOC statement. The default is QSAMOUT.
- disp** Disposition of the output data set. Specify OLD, MOD, NEW, or CAT for NEW,CATLG. The default is NEW.
- dsname** Name of a sequential data set. If specified, any DD statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table.
- fromdate** The earliest creation date, in the form *yyyymmdd*. By default, all objects are copied regardless of their creation date.
- header** If you are copying only one object, specify NO if you do not want a header record. The default is to put a header record at the beginning of the output data set.
- limit** To limit the objects that are copied based on where objects are located, specify one of the following:
- DASD** Only objects that are stored on DASD

OPTICAL Only objects that are stored on optical disks

VOLUME Only objects that are stored on a specified optical disk. You must also specify the volser of the optical disk.

member For a partitioned data set, the member name.

objmask An object name or generic object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is **. In batch mode, this means all objects; in any other mode, this means all objects that have the user's TSO prefix as high-level qualifier.

outds Specification of the sequential output data set (Command Syntax only):

- If a DD or TSO ALLOC statement for QSAMOUT is provided, no input is required.
- If the data set is cataloged specify: *dsname*[(*member*)],,{**OLD|MOD**}.
- If the data set is not cataloged specify: *dsname*[(*member*)],*volser*[,*unit*],{**OLD|MOD**}.
- For a new data set specify: *dsname*[(*member*)],*volser*[,*unit*],{**NEW|CAT**},*space*.

space The number of tracks to be allocated for a new data set. Specify *n* or (*n,m*), where *n* is the number of primary tracks, and *m* is the number of secondary tracks.

todate The latest creation date, in the form *yyyymmdd*.

unit A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

volser On input: if you specified **VOLUME** for *limit*, the volume serial number of the optical disk.

On output: the volume serial number for a new or non-cataloged data set.

Batch Example (MVS)

```
// JOB (acct),'name',Save collection on tape
//S1 EXEC PGM=DITTO
//BACKUP DD DSN='OBJECT.BACKUP',UNIT=CARTR,
//        DISP=(NEW,CATLG)
//SYSPRINT DD SYSOUT=*
$$DITTO OS COLLECTN=SYSU.OAM.CLLCT000,
$$DITTO OUTPUT=BACKUP
$$DITTO EOJ
//
```

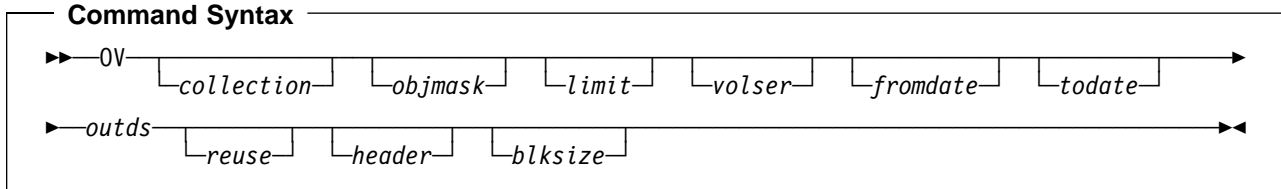
OU (Object Update)—MVS Only

Purpose Browse and update an OAM object at the terminal. This function is available only in full-screen mode under MVS.

For more information about the OU function, use the online help or see “Browsing and Updating OAM Objects in Full-Screen Mode” on page 136.

OV (Object to VSAM)—MVS Only

Purpose Copy one or more OAM objects to a VSAM data set. For more information about using this function, see “Backing up OAM Objects in a Sequential or VSAM Data Set” on page 138.



Batch Syntax

```

$$DITTO OV ...
    
```

Keyword Syntax

```

DITTO $OV ...
    
```

| MVS keywords | VSE keywords | CMS keywords |
|--|--------------|--------------|
| <pre> [COLLECTN={** collection}] [OBJNAME={** objmask}] [LIMIT={DASD OPTICAL VOLUME}] [VOLSER=volser] [FROMDATE=fromdate] [TODATE=todate] [{ OUTPUT={VSAMOUT ddname} DSNOUT=dsname }] [REUSE={NO YES}] [HEADER={YES NO}] [BLKSIZE={4096 blksize}] </pre> | | |

Parameters

- blksize** Block size for the output data set. The maximum depends on the cluster definition; the default is 4 096.
- collection** A collection name. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
The default is all collections; in this case, you must specify LIMIT=VOLUME and a *volser*.
- ddname** Refers to a DD or TSO ALLOC statement. The default is VSAMOUT.
- dsname** Name of a VSAM data set.
- fromdate** The earliest creation date, in the form *yyyymmdd*. By default, all objects are copied regardless of their creation date.
- header** If you are copying only one object, specify NO if you do not want a header record. The default is to put a header record at the beginning of the output data set.
- limit** To limit the objects that are copied based on where objects are located, specify one of the following:
 - DASD** Only objects that are stored on DASD
 - OPTICAL** Only objects that are stored on optical disks
 - VOLUME** Only objects that are stored on a specified optical disk. You must also specify the *volser* of the optical disk.
- objmask** An object name or generic object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers.

In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.

The default is **. In batch mode, this means all objects; in any other mode, this means all objects that have the user's TSO prefix as high-level qualifier.

| | |
|---------------|---|
| <i>outds</i> | Specification of the VSAM output data set (Command Syntax only): <ul style="list-style-type: none"> • If a DD or TSO ALLOC statement for VSAMOUT is provided, no input is required. • If no DD statement is provided, specify: <i>dsname</i>. |
| <i>reuse</i> | Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO. |
| <i>todate</i> | The latest creation date, in the form <i>yyyymmdd</i> . |
| <i>volser</i> | If you specified VOLUME for <i>limit</i> , the volume serial number of the optical disk. |

Keyword Example (MVS)

```

/* REXX */
/* This EXEC saves all OAM objects in a VSAM */
/* data set which were added to a collection */
/* at the current day: */

/* Get date in format YYYYMMDD */
fromdate = DATE(SORTED)
say 'Saving objects from' fromdate

/* create backup copy of all new objects */
"DITTO $OV COLLECTN='SYSU.OAM.CLLCT000',",
"FROMDATE="||fromdate,
"DSNOUT=REUSABLE.ESDS,REUSE=YES"

exit

```

PB (Print Browse)

Purpose Browse (and optionally purge) print output at the terminal. This function is available only in full-screen mode.

Use the SET command to make the print output available for browsing. You do this as follows:

- MVS: Specify a value of SYSOUT=c for the PRINTOUT parameter. The output is then accumulated in a temporary data set and sent to the specified SYSOUT class, when you end the DITTO session.
- VSE and CMS: Specify a value of PRINTER for the PRINTOUT parameter.

The PB function lets you browse the current session output. When browsing, you can enter the PURGE command to purge the print output.

When you leave the Print Browse function without purging the output, DITTO closes the print output and makes it available for printing. Any further print output is again collected and is available for browsing.

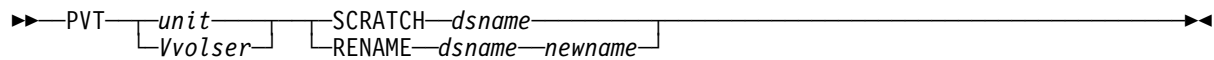
For more information about the PB function, use the online help or see "Browsing Data" on page 32.

PVT (Process VTOC)—VSE and CMS only

Purpose Rename or delete specified VTOC entries. Each renamed or deleted entry is printed in hexadecimal format.

CMS: CMS/DOS is required.

Command Syntax



Batch Syntax

\$\$DITTO PVT ...

Keyword Syntax

DITTO \$PVT ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|---|
| | INPUT ={unit SYSnnn Vvolser} {SCRATCH=dsname RENAME=dsname} [NEWNAME=newname] | INPUT ={unit Vvolser} {SCRATCH=dsname RENAME=dsname} [NEWNAME=newname] |

Parameters

- dsname* The data set name of the VTOC entry to be renamed or deleted.
- newname* The new name, if you are renaming a VTOC entry.
- RENAME Renames an entry.
- SCRATCH Deletes an entry.
- SYSnnn Programmer logical unit to which the disk device is assigned.
- unit* Device number (3 or 4 hexadecimal digits).
- Vvolser* The letter V followed by the serial number of the disk volume.

Batch Example (VSE)

```
// JOB PVT - Process VTOC
// UPSI 1
// EXEC DITTO
$$DITTO PVT INPUT=VSWK3,RENAME=SAM.OUT,
$$DITTO NEWNAME=SAM.OUT.OLD
$$DITTO EOJ
/&
```

Keyword Example (CMS)

```
/* REXX - Process VTOC */
"DITTO $PVT INPUT VSWK3 RENAME SAM.OUT" ,
    "NEWNAME SAM.OUT.OLD"
Exit rc
```


REW (Tape Rewind)

Purpose Rewind a tape to the load point.

Command Syntax

```
▶—REW—intape◀
```

Batch Syntax

```
$$DITTO REW ...
```

Keyword Syntax

```
DITTO $REW ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--|---|---|
| [LABEL=BLP] {INPUT= <i>ddname</i> OUTPUT= <i>ddname</i> } | {INPUT={ <i>unit</i> SYS <i>nnn</i> } OUTPUT={ <i>unit</i> SYS <i>nnn</i> }} | {INPUT= <i>unit</i> OUTPUT= <i>unit</i> } |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
| the first DITTO function that uses the tape. For BLP processing requirements, see
| “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization*
| *Guide*.

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or VOL1) of the tape volume to be mounted. For a volume set, specify up to five volsers in the form (volser1, volser2...).

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form [*yy*]*yy.ddd*.

RUN (Tape Rewind-Unload)

Purpose Rewind a tape and unload it. Under MVS, the tape is closed; the allocation is freed if the tape was dynamically allocated during the current DITTO session.

Command Syntax (MVS and CMS only)

```
▶▶—RUN—intape—————▶▶
```

Command Syntax (VSE only)

```
▶▶—RUN—intape—————▶▶
```

```
└—release—┘
```

Batch Syntax

```
$$DITTO RUN ...
```

Keyword Syntax

```
DITTO $RUN ...
```

| MVS keywords | VSE keywords | CMS keywords |
|---|---|---|
| {INPUT= <i>ddname</i> OUTPUT= <i>ddname</i> } | {INPUT={ <i>unit</i> SYS <i>nnn</i> } OUTPUT={ <i>unit</i> SYS <i>nnn</i> }} | {INPUT= <i>unit</i> OUTPUT= <i>unit</i> } |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit*.

release For an automated tape library, specify YES to release the device from the partition. By default, the device is not released and the next volume (if any) is mounted.

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or V0L1) of the tape volume to be mounted.

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form [*yy*]*yy.ddd*.

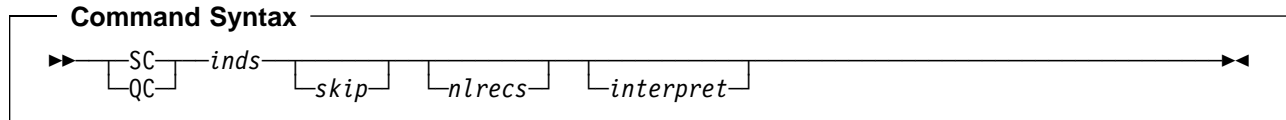
SC, QC (Sequential Data to Card)

Purpose Copy records from a sequential data set to cards. Output is written to:

- MVS: SYSPUNCH
- VSE: SYSPCH
- CMS: The virtual punch address 00D.

The input records must have a record length of 80, 81 (VSE only), or a multiple of 80.

CMS: CMS/DOS is required.



Batch Syntax
 \$\$DITTO SC|QC ...

Keyword Syntax
 DITTO \$\$SC|\$QC ...

| MVS keywords | VSE keywords | CMS keywords |
|--|--|--|
| [{ INPUT={QSAMIN <i>ddname</i> } DSNIN= <i>dsname</i> [(<i>member</i>)] [VOLSERIN= <i>volser</i>] [UNITIN= <i>unit</i>] }] [POSITION={0 <i>skip</i> }] [NLRECS={ALL <i>nlrecs</i> }] [INTERPRET={NO YES}] | [INPUT={ <i>unit</i> SYS <i>nnn</i> <i>vvolser</i> }] [{ FILEIN={SDSKIN <i>dLBL</i> } DSNIN= <i>dsname</i> [VSAMIN={NO YES}] [UCATIN= <i>ucat</i>] }] [POSITION={0 <i>skip</i> }] [NLRECS={ALL <i>nlrecs</i> }] [INTERPRET={NO YES}] | DSNIN= <i>dsname</i> INPUT={ <i>unit</i> <i>vvolser</i> } [POSITION={0 <i>skip</i> }] [NLRECS={ALL <i>nlrecs</i> }] [INTERPRET={NO YES}] |

Parameters

- ddname* Refers to a DD or TSO ALLOC statement. The default is QSAMIN.
- dLBL* Refers to a DLBL statement. The default is SDSKIN.
- dsname* Name of a sequential data set. If specified, any MVS DD or VSE DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table.
- inds* Specification of the sequential input data set. (Command Syntax only.)
- MVS:
- If a DD or TSO ALLOC statement for QSAMIN is provided, no input is required.
 - If the data set is cataloged specify: *dsname*[(*member*)].
 - If the data set is not cataloged specify: *dsname*[(*member*)],*volser*[,*unit*].
- VSE:
- If a DLBL statement for SDSKIN is provided, no input is required.
 - For VSAM-managed SAM specify: *dsname*,YES,[*ucat*].
 - For BAM-managed SAM specify: *dsname*,NO,{*unit*|*vvolser*}.

CMS:

- Specify: *dsname*,{*unit*|**V***volser*}.

| | |
|------------------|--|
| <i>interpret</i> | Interpret the output. If you specify YES, the output device must be a card punch with the card print feature installed. The default is NO. |
| <i>member</i> | For a partitioned data set, the member name. |
| <i>nlrecs</i> | Number of cards to be punched or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are punched. |
| <i>skip</i> | Number of card images to be skipped from the beginning of the data set. The default is 0. |
| <i>SYSnnn</i> | Programmer logical unit to which the input device is assigned. |
| <i>ucat</i> | For VSAM-managed SAM, the data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number for a non-cataloged data set. |
| <i>vsamin</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the input volume. |

Batch Example (MVS)

```
//SC JOB (acct),'name' QSAM to SYSPCH
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.HOGB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//S1 EXEC DTOBAT
//SYSPUNCH DD SYSOUT=*
$$DITTO QC DSNIN=SYS1.PARMLIB(IEASYS00),
$$DITTO NLRECS=ALL
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB SC - SAM to SYSPCH
// UPSI 1
// EXEC DITTO
$$DITTO QC DSNIN=MYJOB.JCL,
$$DITTO INPUT=VSYSWK3,NLRECS=15
$$DITTO EOJ
/*
/&
```

Keyword Example (CMS)

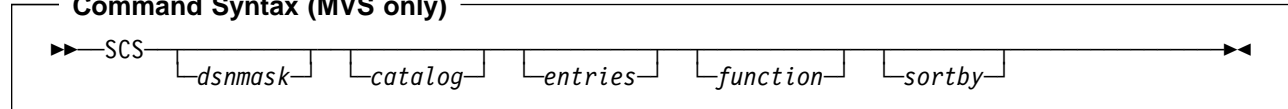
```
/* REXX EXEC to send a job to a VSE machine */
/* spool virtual punch to VSE machine */
address CP "SPOOL 00D TO VSETEST NOCONT"
/* run command mode DITTO SAM to PUNCH */
"DITTO $SC DSNIN=MYJOB.JCL,",
"INPUT=VSYSWK3,NLRECS=15"
address CP "CLOSE 00D" /* close punch */
address CP "SPOOL 00D TO *" /* restore dest.*/
exit
```

SCS (Catalog Services)—VSE and MVS Only

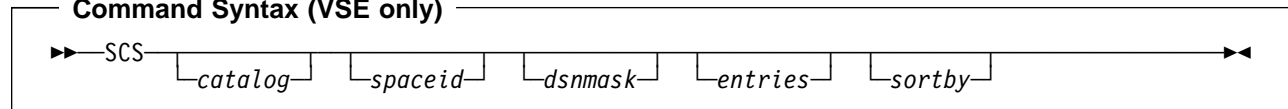
Purpose Work with a list of catalog entries. In full-screen mode, you can perform various DITTO and IDCAMS functions. In command mode under MVS, you can invoke the full-screen mode SCS function. In any other case, you can only print a list of catalog entries.

For more information about using this function in full-screen mode, see Chapter 5, “Processing Catalog Entries” on page 51.

Command Syntax (MVS only)



Command Syntax (VSE only)



Batch Syntax

```
$$DITTO SCS ...
```

Keyword Syntax

```
DITTO $SCS ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--|--|--------------|
| [DSNAME={** <i>dsnmask</i> }] [CATALOG= <i>catalog</i>] [ENTRIES={ALL ALLVSAM ALIAS AIX CLUSTER DATA GDG INDEX NONVSAM OAM PGSPC PATH UCAT}] [FUNCTION={LIST ALTER DEFINE DELETE PRINT}] [SORTBY={NAME DATE EDATE FREESP}] | [{CATALOG= <i>catalog</i> FILEIN= <i>dlbl</i> }] [VOLSER= <i>spaceid</i>] [DSNAME={** <i>dsnmask</i> }] [ENTRIES={ALL ALLVSAM AIX CLUSTER DATA INDEX NONVSAM PATH SPACE UCAT}] [PWDIN= <i>pwdin</i>] [SORTBY={NAME DATE EDATE FREESP EXTENT}] | |

Parameters

catalog Name of a catalog.

MVS: by default, the system catalog search order is used.

VSE: the default is the master catalog.

dlbl Refers to a DLBL statement.

dsnmask A generic data set name, or search pattern. Within the data set name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers. The default is **.

Under MVS in any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes. For example, the default of ** lists all entries in batch mode; in any other mode, it lists all entries that have the user's TSO prefix as high-level qualifier.

entries The entries you want listed:

ALL All catalog entries (the default)
ALLVSAM All VSAM catalog entries
ALIAS MVS: Alias entries
AIX Alternate index entries
CLUSTER Cluster entries

DATA VSAM data entries
GDG MVS: Generation data group entries
INDEX VSAM index entries
NONVSAM NonVSAM entries
OAM MVS: OAM collection entries
PATH VSAM path entries
PGSPC MVS: Pagespace entries
SPACE VSE: VSAM data space entries
UCAT User catalog entries.

function The command that you want to execute:

LIST Display information about the selected catalog entries
ALTER Change information in the catalog about the specified entry
DEFINE Create new catalog entry
DELETE Delete the specified catalog entry and its related data
PRINT Print information about the selected catalog entries.

| MVS: in batch mode, DITTO assumes that FUNCTION=PRINT is specified. The FUNCTION
| keyword is ignored if specified.

pwdin Password of the catalog.

sortby One of the following:

NAME Sorts by data set name (the default)
DATE Sorts by creation date
EDATE Sorts by expiration date
FREESP Sorts by free space
EXTENT VSE: sorts the contents of a VSAM data space by extent.

spaceid The serial number of the volume that contains a VSAM space, if you want to list only the data sets within that data space.

Batch Example (MVS)

```

//SCS JOB (acct),'name' Catalog Services
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DELETE EXEC PGM=IEFBR14
//FILE DD DSN=KEISTEW.SCS.LISTING,
// DISP=(MOD,DELETE),
// SPACE=(TRK,(30,10)),UNIT=SYSDA,
// RECFM=FBA,LRECL=133
//*
//SCS00 EXEC DTOBAT,REGION=6M
//SYSPRINT DD DSN=KEISTEW.SCS.LISTING,
// DISP=(,CATLG),
// SPACE=(TRK,(30,10)),UNIT=SYSDA,
// RECFM=FBA,LRECL=133
//SYSIN DD *
$DITTO SCS
$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB SCS - Catalog Services
// UPSI 1
// EXEC DITTO
$$DITTO SCS CATALOG=VSAM.MASTER.CATALOG,
$DITTO DSNAME=**
$DITTO EOJ
/&

```

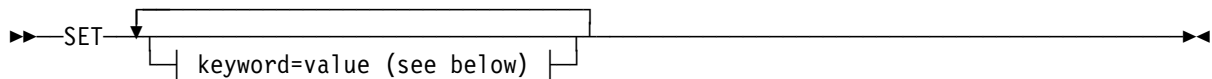
SET (Set Processing Options)

Purpose List the processing options that are currently in effect, change one or more options, or reset to the installation default.

If you use the SET function from the VSE or MVS system console, within a REXX/VSE procedure, or within a batch job, your changes affect only the current session or batch job. In any other case, your changes are saved in the user profile and are available for further DITTO sessions.

To change the installation defaults that apply to *all* users and batch jobs, see the *DITTO/ESA Installation and Customization Guide*.

Command Syntax



Batch Syntax

\$\$DITTO SET ...

Keyword Syntax

DITTO \$SET ...

| MVS keywords | VSE keywords | CMS keywords |
|--|--|--|
| [RESET] [ASCII={NO IN OUT BOTH}] [CMDSYNTAX={CMS DITTO ISPF}] [CYLHD={ABSOLUTE RELATIVE}] [DATAHDR={YES NO}] [DBCSPRT={OFF 3200 SOSI}] [DUMP={UPDOWN ACROSS}] [EOD= <i>delimiter</i>] [HEADERPG={YES NO}] [LANGUAGE={ENGLISH <i>language</i> }] [PAD={OFF ON <i>c</i> }] [PAGESIZE={60 <i>nnn</i> }] [PAGESKIP={NO YES}] [PRINTLEN={132 80}] [PRINTOUT={SYSPRINT SYSOUT= <i>c</i> TERMINAL REXX}] [PRTRAN={ON OFF KN}] [RECLIMIT={{1,*} (<i>n,m</i>)}] [TAPEASGN={TEMP PERM}] [TAPELBL={SL AL AL4}] | [RESET] [ASCII={NO IN OUT BOTH}] [CMDSYNTAX={CMS DITTO ISPF}] [DATAHDR={YES NO}] [DBCSPRT={OFF 3200 SOSI}] [DUMP={UPDOWN ACROSS}] [EOD= <i>delimiter</i>] [HEADERPG={YES NO}] [LANGUAGE={ENGLISH <i>language</i> }] [PAD={OFF ON <i>c</i> }] [PAGESIZE={STDOPT <i>nnn</i> }] [PAGESKIP={NO YES}] [PRINTLEN={132 80}] [PRINTOUT={PRINTER TERMINAL REXX}] [PRTRAN={ON OFF KN}] [RECLIMIT={{1,*} (<i>n,m</i>)}] [TAPEASGN={TEMP PERM}] [TAPELBL={SL AL AL4}] | [RESET] [ASCII={NO IN OUT BOTH}] [CMDSYNTAX={CMS DITTO ISPF}] [DATAHDR={YES NO}] [DBCSPRT={OFF 3200 SOSI}] [DUMP={UPDOWN ACROSS}] [EOD= <i>delimiter</i>] [HEADERPG={YES NO}] [LANGUAGE={ENGLISH <i>language</i> }] [PAD={OFF ON <i>c</i> }] [PAGESIZE={60 <i>nnn</i> }] [PAGESKIP={NO YES}] [PRINTLEN={132 80}] [PRINTOUT={PRINTER TERMINAL REXX}] [PRTRAN={ON OFF KN}] [RECLIMIT={{1,*} (<i>n,m</i>)}] [TAPELBL={SL AL AL4}] |

Keywords

Enter SET with no parameters to display the current settings.

RESET Returns all options to the installation default values (described in the *DITTO/ESA Installation and Customization Guide*).

ASCII=NO|IN|OUT|BOTH

- ASCII=NO Does not translate data. This is the default.
- ASCII=IN Translates tape input from ASCII format to EBCDIC format.
- ASCII=OUT Translates tape output from EBCDIC format to ASCII format.
- ASCII=BOTH Translates tape input from ASCII format to EBCDIC format, and translates tape output from EBCDIC format to ASCII format.

This parameter applies to all tape input and output functions except: INT, TB, TFA, TFD, TLB, TLT, TTC, and TU. For the TLB function, ASCII translation is automatically performed when needed. For the INT and TB functions, you can specify ASCII input or output when invoking the function. For the other functions, translation is not supported by DITTO.

CMDSYNTAX=CMS|DITTO|ISPF

Specifies the syntax of the browse and edit LOCATE command and the edit CHANGE command:

CMDSYNTAX=CMS The LOCATE command is a synonym for the FIND command, and has the same syntax as in DITTO/ESA Release 2 and earlier:

LOCATE *string*

The CHANGE command has syntax similar to the CMS editor CHANGE command:

CHANGE */old/new/* Changes the first occurrence in the current record only.

CHANGE */old/new/ n* In multiple-record mode: changes the first occurrence in the current record, and the first occurrence in each of the following *n-1* records.

In single-record mode: changes the first occurrence in the current record only.

CHANGE */old/new/ ** In multiple-record mode: changes the first occurrence in the current record, and the first occurrence in each subsequent record.

In single-record mode: changes the first occurrence in the current record only.

CHANGE */old/new/ ALL* Same as above.

CHANGE */old/new/ * n* In multiple-record mode: changes the first *n* occurrences in the current record, and the first *n* occurrences in each subsequent record.

In single-record mode: changes the first *n* occurrences in the current record only.

CHANGE */old/new/ 1 ** Changes all occurrences in the current record.

CHANGE */old/new/ * ** In multiple-record mode: changes all occurrences in the current record and in all subsequent records.

In single-record mode: changes all occurrences in the current record only.

CHANGE */old/new/ ALL ALL* Same as above.

The first non-blank delimiter is used as a delimiter (in the examples above, the / character). If *old* is a null string, the *new* string is inserted at the current column position.

The *old* string is case sensitive. For example:

CHANGE */before/after/*

changes only “before,” not “BEFORE” or “Before.”

SET

If CAPS ON or CASE UPPER is in effect, the CHANGE command behaves as if the *new* string was specified in all uppercase. For details, see “Translating Input to Uppercase” on page 46.

This is the default under VM.

CMDSYNTAX=DITTO The LOCATE command is a synonym for the FIND command, and has the same syntax as in DITTO/ESA Release 2 and earlier:

LOCATE *string*

The CHANGE command has the same syntax as in DITTO/ESA Release 2 and earlier:

CHANGE */old/new/* In multiple-record mode: changes the next occurrence.

In single-record mode: changes the next occurrence in the current record only.

CHANGE */old/new/ n* In multiple-record mode: changes the next *n* occurrences.

In single-record mode: changes the next *n* occurrences in the current record only.

CHANGE */old/new/ ** In multiple-record mode: changes the next occurrence and all subsequent occurrences.

In single-record mode: changes all occurrences in the current record.

CHANGE */old/new/ ALL* Same as above.

The first non-blank delimiter is used as a delimiter (in the examples above, the / character). If *old* is a null string, the *new* string is inserted at the current column position.

The *old* string is case sensitive.

If CAPS ON or CASE UPPER is in effect, the CHANGE command behaves as if the *new* string was specified in all uppercase. For details, see “Translating Input to Uppercase” on page 46.

This is the default under VSE.

CMDSYNTAX=ISPF The LOCATE command has syntax similar to the ISPF editor LOCATE command:

LOCATE *line_number*

Notes:

1. This form of LOCATE is supported in the following functions:

Browse functions APB, FB (except when in single-record mode), LB (except when browsing a member stored in string format such as a PHASE, or when in single-record mode), PB, and SPB

Edit functions DTE (single-record mode only), VE (multiple-record mode only), and LE

List functions DLA, DVT, LDL, SCS, SPL, SPS, and VDL

2. In the APB function, the syntax is:

```
LOCATE page_number
```

The CHANGE command has syntax similar to the ISPF editor CHANGE command:

CHANGE *old new* In multiple-record mode: Changes the next occurrence.
 In single-record mode: Changes the next occurrence in the current record only.

CHANGE *old new ** In multiple-record mode: changes all occurrences in all records.
 In single-record mode: changes all occurrences in the current record.

CHANGE *old new ALL* Same as above.

The *old* string can be:

- A character string without blanks or commas. Case (uppercase or lowercase) is ignored.
- A character string enclosed in single quotation marks. It can contain blanks and commas. The string must match exactly (including case).
- X followed by a hexadecimal string enclosed in single quotation marks. (for example, X'C1C2').

The *new* string can be:

- A character string without blanks or commas. Case is respected and retained, unless CAPS ON or CASE UPPER is in effect, in which case the CHANGE command behaves as if the *new* string was specified in all uppercase. For details, see “Translating Input to Uppercase” on page 46.
- A character string enclosed in single quotation marks. It can contain blanks and commas. Case is respected and retained, unless CASE UPPER is in effect, in which case the CHANGE command behaves as if the *new* string was specified in all uppercase. For details, see “Translating Input to Uppercase” on page 46.
- X followed by a hexadecimal string enclosed in single quotation marks. (for example, X'C1C2'). If CASE UPPER is in effect, the CHANGE command behaves as if the hexadecimal string for the equivalent uppercase characters was specified. For details, see “Translating Input to Uppercase” on page 46.

If *old* is a null string (' '), the *new* string is inserted at the current column position.

This is the default under MVS.

For more information about using the LOCATE command, see “Locating a Specific Record” on page 33 and “Locating Specific Data” on page 37 (browse functions), or “Locating a Specific Record” on page 41 (edit functions). For more information about using the CHANGE command, see “Locating and Changing a String” on page 43.

Note: In single-record mode, the CHANGE command affects only the current record.

SET

CYLHD=ABSOLUTE|RELATIVE

MVS: Defines the way you specify the location of a disk data set in disk functions. DITTO interprets cylinder-head and track-number values as:

- Actual physical addresses, if CYLHD=ABSOLUTE
- Relative to the start of the data set, if CYLHD=RELATIVE.

DATAHDR=YES|NO

Specifies whether record header information (for example, record number and length) is included in character-format print output. Specify DATAHDR=NO to left-justify the data in the print output, without header information.

Note: Because card output has a fixed length of 80 characters, the Card Print (CP) function ignores the DATAHDR parameter and never prints a data header.

DBCSPRT=OFF|3200|SOSI

Specifies how DBCS fields are formatted for printing:

DBCSPRT=OFF No special formatting is performed; EBCDIC format is used.

DBCSPRT=3200 DBCS fields are formatted for the IBM 3200 printer. DITTO puts the function command X'0A42' at the start and X'0A41' at the end of each DBCS field. For mixed format fields, DITTO adds a blank before X'0A42' and after X'0A41' to ensure that the output line is correctly positioned.

DBCSPRT=SOSI DBCS fields are formatted for the IBM 5550 terminal. DITTO puts the shift out (SO) character X'0E' at the start and the shift in (SI) character X'0F' at the end of each DBCS field. For mixed format fields, DITTO adds a blank before X'0E' and after X'0F' to ensure that the output line is correctly positioned.

This option affects only character output produced by DITTO print commands. It affects only data that has been identified as DBCS data by the FMT command. For more information about the FMT command, see "FMT (Set DBCS Format)" on page 192.

DUMP=UPDOWN|ACROSS

Specifies the format of hexadecimal output (for example, from the Tape Print function with FORMAT=HEX). For an example of each format, see "Formats for Displaying or Printing Data" on page 4.

EOD=*delimiter*

End-of-data delimiter for card and tape input. You can specify a delimiter of up to 8 characters.

For character data, enclose the string in single quotation marks if it contains blanks, commas, or lowercase letters. For hexadecimal data, enter an X followed by the string enclosed in single quotation marks (for example, X'04').

Card input: All functions that process card input continue until the end of the card deck, or a card that begins with the EOD delimiter. In batch mode, the default delimiter is a slash followed by an asterisk (/ *); in any other mode, the default delimiter is a blank card (80 blanks).

Tape input: For some tape functions, you specify the number of files to process. If you specify EOD, files are copied until the first record that begins with the EOD delimiter. There is no default delimiter for tape input.

To reset the EOD delimiter to the default value, enter EOD=".

HEADERPG=YES|NO

Specifies whether a header page (a title page with DITTO/ESA on it) is included in print output. A header page does not appear on output routed to the terminal, even if HEADERPG=YES is set.

LANGUAGE=ENGLISH|FRENCH|GERMAN|ITALIAN|JAPANESE|PORTUGUESE|SPANISH|OTHER

Specifies the language of DITTO message text and panel text. The possible values depend on which translated message text you installed.

For instructions on installing translated messages and panels, see the *DITTO/ESA Installation and Customization Guide*.

PAD=OFF|ON|c

Specifies whether records are padded or truncated while being copied.

PAD=OFF Records are not padded or truncated.

PAD=ON Records are modified, depending on the output record format. For fixed-format output, input records that do not fit are truncated or padded with blanks. For variable or undefined output, trailing blanks are stripped from the input record. (For reasonable results, do not try to deblock and pad in one run.)

PAD=c The same as PAD=ON except that the specified character *c* is used for padding and truncating instead of blanks.

PAGESIZE=60|nnn (MVS and CMS)

= **STDOPT|nnn (VSE only)**

Specifies the number of lines printed on each page of print output. This can be a value from 1 to 999. The default is 60.

VSE: specify STDOPT (the default) to use the page size specified in the standard job control option.

PAGESKIP=NO|YES

Specifies whether print output from each function begins on a new page. Use PAGESKIP=YES if you want to skip to a new page for each function.

PRINTLEN=132|80

Specifies whether the line length of print output is suitable for a terminal (80 characters) or for a printer (132 characters).

PRINTOUT=SYSPRINT|SYSOUT=c|TERMINAL|REXX (MVS only)

= **PRINTER|TERMINAL|REXX (VSE and CMS)**

Specifies where print output is sent, in any mode except batch mode.

MVS: you can send output to the current SYSPRINT allocation, send it to a specific SYSOUT class *c* (where *c* can be A-Z or 0-9), display it on a TSO terminal, or write it to the REXX stem variable DITTO..

VSE and CMS: you can send output to the printer, display it on your terminal, or write it to the REXX stem variable DITTO..

For more information about these options, see "DITTO Print Output" on page 26.

PRTRANS=ON|OFF|KN

PRTRANS=ON Specifies that non-printable characters are translated to blanks using a translate table. This may make printing faster.

PRTRANS=OFF Specifies that no translation will take place. Use PRTRANS=OFF to support special print chains.

PRTRANS=KN Specifies that non-printable characters are translated to blanks using a translate table, and all message text is translated to uppercase. Use this value if your printer supports the Latin alphabet (used for languages such as English) only in uppercase. For example, if your printer supports the Cyrillic alphabet (used for

SET

languages such as Russian) or Katakana characters (used in Japanese), the printer may not support lowercase Latin letters.

| For instructions on altering the print translate table, see the *DITTO/ESA Installation and Customization Guide*.

RECLIMIT=(1,*)(n,m)

Limits print output for each record (or block of a library member or OAM object) to the data from columns (bytes) *n* through *m*. An asterisk (*) indicates the end of the record.

TAPEASGN=TEMP|PERM

VSE: specifies whether tapes are assigned temporarily (for the duration of one function only) or permanently (for the duration of the DITTO session).

This parameter affects full-screen mode and line mode sessions only. In batch mode, tapes are always assigned for the duration of the batch job. In command mode, you leave DITTO immediately after a command is executed (so the duration of a function and the duration of a session are the same).

TAPELBL=SL|AL|AL4

| Specifies the type of labels that are created:

TAPELBL=SL EBCDIC labels

| TAPELBL=AL ANSI Version 3 labels ("Version 3" refers to ANSI X3.27–1978, ISO 1001–1979,
| and FIPS 79 standards)

| TAPELBL=AL4 ANSI Version 4 labels ("Version 4" refers to ANSI X3.27–1987 level 4 and
| ISO 1001–1986(E) standards)

| For detailed information about ANSI tape labels, see *DFSMS/MVS Using Magnetic Tapes*,
| SC26-4923.

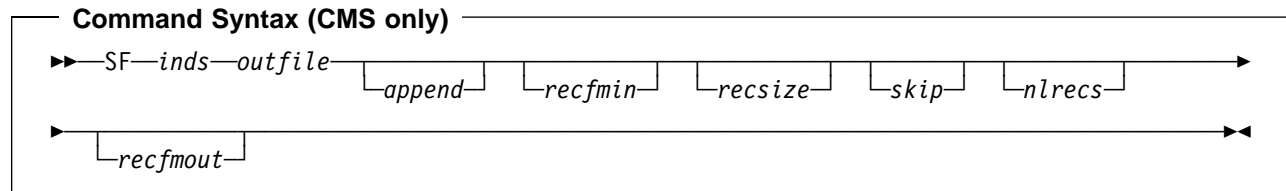
MVS: This parameter affects only the INT function.

SF (Sequential to CMS File)—CMS and CMS/VSE Mode Only

Purpose Copy records from a sequential data set to a CMS file.

You can use this function:

- Under CMS, if you have CMS/DOS.
- In CMS/VSE mode, to copy data from a VSE system to CMS.



| Batch Syntax (CMS only) | Keyword Syntax (CMS only) |
|-------------------------|---------------------------|
| \$DITTO SF ... | DITTO \$SF ... |

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--------------|---|
| | | DSNIN= <i>dsname</i> INPUT={ <i>unit</i> <i>Vvolser</i> } [RECFMIN={ <i>U</i> <i>V</i> <i>VB</i> <i>F</i> <i>FB</i> }] [RECLSIZE= <i>reclsize</i>] [POSITION={ <i>0</i> <i>skip</i> }] [NLRECS={ <i>ALL</i> <i>nlrecl</i> }] OUTFILE= <i>'outfile'</i> [APPEND={ <i>NO</i> <i>YES</i> }] [RECFMOUT={ <i>V</i> <i>U</i> <i>F</i> }] |

Parameters

- append* Specify YES to add output to the end of an existing CMS file. By default, any existing CMS file with the specified file ID is replaced.
- dsname* Name of a sequential data set.
- inds* Specification of the sequential input data set. (Command Syntax only.) Specify: *dsname*,{*unit*|*Vvolser*}.
- nlrecl* Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.
- outfile* Specifies the CMS output file identifier: *filename filetype [filemode]*, enclosed in single quotes. The default filemode is A.
- recfmin* Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:
- B** Blocked
 - F** Fixed length
 - U** Undefined length
 - V** Variable length
- recfmout* Record format for the output:
- F** Checks that all input records are fixed-length, and creates a fixed format output file.
 - V** or **U** Processes the input records without a length check, and creates a variable-format output file (the default).

SF

- recsize* Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.
- skip* Number of logical records to be skipped from the beginning of the data set. The default is 0.
- unit* Device number (3 or 4 hexadecimal digits).
- Vvolser* The letter V followed by the serial number of the input volume.

Keyword Example (CMS)

```
/* run command mode DITTO SAM to CMS file */
"DITTO $SF DSNIN=MYJOB.DATA,INPUT=VSYSWK3,",
  "RECFMIN=VB,OUTFILE='MYDATA DATA'",
  "RECFMOUT=V,APPEND=NO"
address CMS "FINIS MYDATA DATA"
exit
```


SFA (Print SYSLST File Type A)—VSE Only

Purpose Print a SYSLST disk file that contains CCW command codes.

Command Syntax

```

▶—SFA—inds—┌──recfmin──┐┌──recsize──┐┌──skip──┐┌──nlrecs──┐

```

Batch Syntax

```
$$$DITTO SFA ...
```

Keyword Syntax

```
DITTO $$FA ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | <pre> [INPUT={unit SYSnnn Vvolser}] [FILEIN={SDSKIN dlbl} DSNIN=dsname VSAMIN={NO YES}] [UCATIN=ucatl]] [RECFMIN={U V VB F FB}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] </pre> | |

Parameters

- dlbl* Refers to a DLBL statement. The default is SDSKIN.
- dsname* Name of a sequential data set. If specified, any DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table.
- inds* Specification of the sequential input data set. (Command Syntax only.)
- If a DLBL statement for SDSKIN is provided, no input is required.
 - For VSAM-managed SAM specify: *dsname*, **YES**, [*ucatl*].
 - For BAM-managed SAM specify: *dsname*, **NO**, {*unit*|*Vvolser*}.
- nlrecs* Number of records to be printed or ALL. The maximum number is 99999999. If you specify ALL or omit the parameter, all the remaining records are printed.
- recfmin* Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:
- B** Blocked
F Fixed length
U Undefined length
V Variable length
- recsize* Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.
- skip* Number of logical records to be skipped from the beginning of the data set. The default is 0.
- SYSnnn* Programmer logical unit to which the input device is assigned.
- ucatl* For VSAM-managed SAM, the data set name of the user catalog.
- unit* Device number (3 or 4 hexadecimal digits).
- vsamin* Specify YES for VSAM-managed SAM. The default is NO.
- Vvolser* The letter V followed by the serial number of the input volume.

SFD (Print SYSLST File Type D)—VSE Only

Purpose Print a SYSLST disk file that contains ASA control characters. (Files produced by VSE job control contain ASA control characters.) To print a SYSLST disk file that contains CCW command codes, use SFA.

ASA control characters are translated to their corresponding CCW command codes. For any undefined character, a space and single line are printed.

Command Syntax

```

▶—SFD—inds—┌──recfmin──┐┌──recsize──┐┌──skip──┐┌──nlrecs──┐

```

Batch Syntax

```
$$DITTO SFD ...
```

Keyword Syntax

```
DITTO $SFD ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | <pre> [INPUT={unit SYSnnn Vvolser}] [{ FILEIN={SDSKIN dlbl} DSNIN=dsname [VSAMIN={NO YES}] [UCATIN=ucat]]}] [RECFMIN={U V VB F FB}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] </pre> | |

Parameters

- dlbl* Refers to a DLBL statement. The default is SDSKIN.
- dsname* Name of a sequential data set. If specified, any DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table.
- inds* Specification of the sequential input data set. (Command Syntax only.)
- If a DLBL statement for SDSKIN is provided, no input is required.
 - For VSAM-managed SAM specify: *dsname*, **YES**, [*ucat*].
 - For BAM-managed SAM specify: *dsname*, **NO**, {*unit*|**V***volser*}.
- nlrecs* Number of records to be printed or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are printed.
- recfmin* Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:
- B** Blocked
 - F** Fixed length
 - U** Undefined length
 - V** Variable length
- recsize* Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.
- skip* Number of logical records to be skipped from the beginning of the data set. The default is 0.
- SYSnnn* Programmer logical unit to which the input device is assigned.
- ucat* For VSAM-managed SAM, the data set name of the user catalog.

- unit* Device number (3 or 4 hexadecimal digits).
- vsamin* Specify YES for VSAM-managed SAM. The default is NO.
- Vvolser* The letter V followed by the serial number of the input volume.

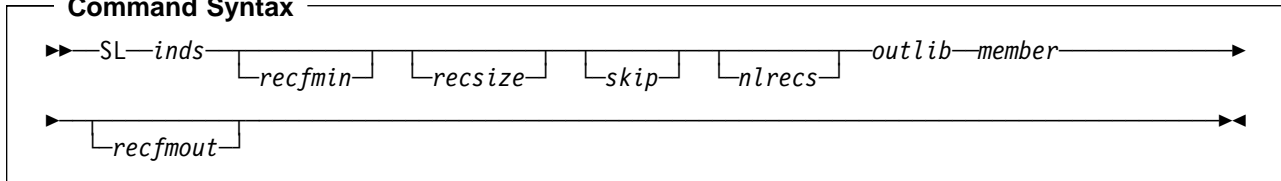
Batch Example (VSE)

```
// JOB SFD - Print SYSLST File Type D
// UPSI 1
// DLBL DITLIST, 'DITTO.SYSLST.FILE'
// EXEC DITTO
$$DITTO SFD INPUT=VSZYWK1,FILEIN=DITLIST,
$$DITTO RECFMIN=F,RECSIZE=121
$$DITTO EOJ
/&
```

SL (Sequential Data to Library Member)—VSE Only

Purpose Copy records from a sequential data set to a library member.

Command Syntax



Batch Syntax

\$\$\$DITTO SL ...

Keyword Syntax

DITTO \$SL ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | [INPUT={unit SYSnnn Vvolser}] [{ FILEIN={SDSKIN dlbl} DSNIN=dsname [VSAMIN={NO YES}] [UCATIN=ucat]]] [RECFMIN={U V VB F FB}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] LIBOUT=lib MEMBEROUT=member [RECFMOUT={F S}] | |

Parameters

- dlbl* Refers to a DLBL statement. The default is SDSKIN.
- dsname* Name of a sequential data set. If specified, any DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table.
- inds* Specification of the sequential input data set. (Command Syntax only.)
- If a DLBL statement for SDSKIN is provided, no input is required.
 - For VSAM-managed SAM specify: *dsname*, **YES**, [*ucat*].
 - For BAM-managed SAM specify: *dsname*, **NO**, {*unit*|**V***volser*}.
- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- member* Name of the library member in the form *name.type*.
- nlrecs* Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.
- outlib* Specification of the output library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary*, *libdsn*, *volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary*, *libdsn*, [, [*ucat*].
- recfmin* Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:
- B** Blocked
 - F** Fixed length
 - U** Undefined length

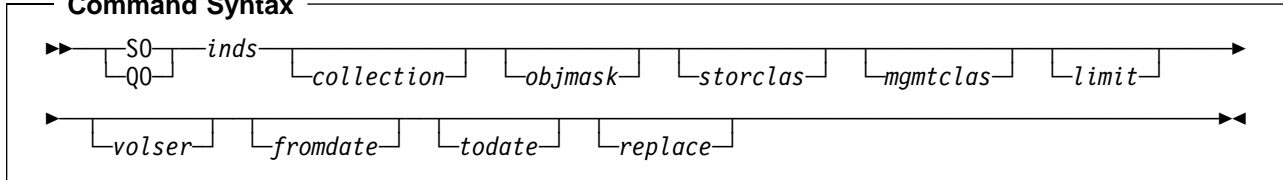
| | |
|-----------------|---|
| | V Variable length |
| <i>recfmout</i> | Record format for the output: F Fixed-length, 80 (the default) S String format (always used if the member type is PHASE). |
| <i>reclsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>reclsize</i> is required. |
| <i>skip</i> | Number of logical records to be skipped from the beginning of the data set. The default is 0. |
| <i>SYSnnn</i> | Programmer logical unit to which the input device is assigned. |
| <i>ucat</i> | For VSAM-managed SAM or library space, the data set name of the user catalog. |
| <i>unit</i> | Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number of the device where the library is located. |
| <i>vsamin</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the input volume. |

SO, QO (Sequential Data to Object)—MVS Only

Purpose Restore one or more OAM objects that were backed up in a sequential data set, or create a new object from a sequential data set. For more information about using this function, see “Restoring OAM Objects from a Sequential or VSAM Data Set” on page 139.

For compatibility with previous releases, you can specify either SO or QO.

Command Syntax



Batch Syntax

\$\$\$DITTO SO|QO ...

Keyword Syntax

DITTO \$\$SO|\$\$QO ...

| MVS keywords | VSE keywords | CMS keywords |
|--|--------------|--------------|
| <pre>[{ INPUT={QSAMIN ddname} DSNIN=dsname[(member)] [VOLSERIN=volser] [UNITIN=unit] }] [COLLECTN={** collection}] [OBJNAME={** objmask}] [STORCLAS=storclas] [MGMTCLAS=mgmtclas] [LIMIT={DASD OPTICAL VOLUME}] [VOLSER=volser] [FROMDATE=fromdate] [TODATE=todate] [REPLACE={NO YES}]</pre> | | |

Parameters

- collection* If the data set contains one object with a header, you can optionally specify a collection name to override the collection name in the header.
- If the data set contains one object with no header (or, equivalently, it contains data that was not previously an object), you must specify the collection where the object will be stored.
- If the data set contains one or more objects that were stored during a multiple-object backup, you can optionally specify a collection name to restore only objects from the specified collection.
- In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
- ddname* Refers to a DD or TSO ALLOC statement. The default is QSAMIN.
- dsname* Name of a sequential data set. If specified, any DD statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table.
- fromdate* The earliest creation date, in the form *yyyymmdd*. By default, all objects are copied regardless of their creation date.
- inds* Specification of the sequential input data set (Command Syntax only):
- If a DD or TSO ALLOC statement for QSAMIN is provided, no input is required.
 - If the data set is cataloged specify: *dsname[(member)]*.
 - If the data set is not cataloged specify: *dsname[(member)],volser[,unit]*.

| | |
|-----------------|---|
| <i>limit</i> | To limit the objects that are copied based on where the objects were located (before they were backed up to a data set), specify one of the following: DASD Only objects that were stored on DASD OPTICAL Only objects that were stored on optical disks VOLUME Only objects that were stored on a specified optical disk. You must also specify the volser of the optical disk. |
| <i>member</i> | For a partitioned data set, the member name. |
| <i>mgmtclas</i> | Name of an SMS management class. |
| <i>objmask</i> | If the data set contains one object with a header, you can optionally specify an object name to override the object name in the header. If the data set contains one object with no header (or, equivalently, it contains data that was not previously an object), you must specify a name for the object. If the data set contains one or more objects that were stored during a multiple-object backup, you can optionally specify an object name or generic object name to limit the objects that are restored based on object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) with any number of qualifiers. In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes. |
| <i>replace</i> | Specify YES to replace any existing objects with the same name in the collection. By default, the object is not copied. |
| <i>storclas</i> | Name of an SMS storage class. |
| <i>todate</i> | The latest creation date, in the form <i>yyyymmdd</i> . |
| <i>unit</i> | A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. |
| <i>volser</i> | On input: the volume serial number for a non-cataloged data set. On output: if you specified VOLUME for <i>limit</i> , the volume serial number of the optical disk. |

Batch Example (MVS)

```
//SO JOB (acct),'name' Restore Objects
//DITTO EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//OBJBKUP DD DSN='DTT.OBJECT.BKUP',
//          VOL=SER=DIT010,UNIT=CARTR,
//          DISP=OLD
$$DITTO SO INPUT=OBJBKUP,OBJNAME=**,
$$DITTO COLLECTN=SYSU.OAM.CLLCT000,
$$DITTO LIMIT=VOLUME,VOLSER=OVOL03
$$DITTO EOJ
//
```

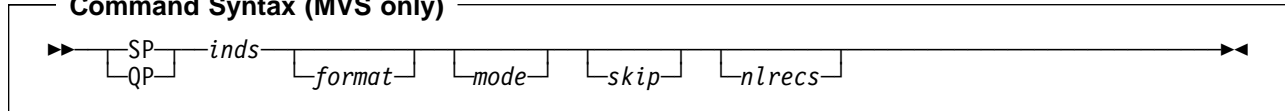
SP, QP (Sequential Data Print)

Purpose Print records from a sequential data set in character or hexadecimal format.

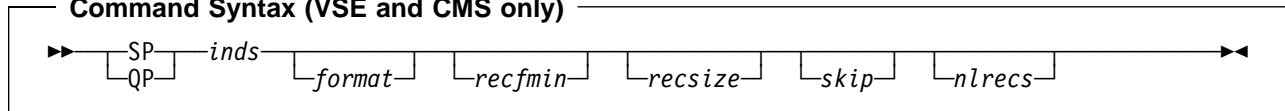
CMS: CMS/DOS is required.

For compatibility with previous releases, you can specify either SP or QP.

Command Syntax (MVS only)



Command Syntax (VSE and CMS only)



Batch Syntax

\$\$\$DITTO SP|QP ...

Keyword Syntax

DITTO \$\$\$P|\$QP ...

| MVS keywords | VSE keywords | CMS keywords |
|---|--|--|
| [{ INPUT={QSAMIN <i>ddname</i> } DSNIN= <i>dsname</i> [(<i>member</i>)] [VOLSERIN= <i>volser</i>] [UNITIN= <i>unit</i>] }] [FORMAT={CHAR HEX}] [MODE=U] [POSITION={0 <i>skip</i> }] [NLRECS={ALL <i>nlrecs</i> }] | [INPUT={ <i>unit</i> SYS <i>nnn</i> <i>vvolser</i> }] [{ FILEIN={SDSKIN <i>dlbl</i> } DSNIN= <i>dsname</i> [VSAMIN={NO YES}] [UCATIN= <i>ucat</i>] }] [FORMAT={CHAR HEX}] [RECFMIN={U V VB F FB}] [RECSIZE= <i>recsize</i>] [POSITION={0 <i>skip</i> }] [NLRECS={ALL <i>nlrecs</i> }] | DSNIN= <i>dsname</i> INPUT={ <i>unit</i> <i>vvolser</i> } [FORMAT={CHAR HEX}] [RECFMIN={U V VB F FB}] [RECSIZE= <i>recsize</i>] [POSITION={0 <i>skip</i> }] [NLRECS={ALL <i>nlrecs</i> }] |

Parameters

ddname Refers to a DD or TSO ALLOC statement. The default is QSAMIN.

dlbl Refers to a DLBL statement. The default is SDSKIN.

dsname Name of a sequential data set. If specified, any MVS DD or VSE DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table.

format The format of the output:
CHAR Character format (the default)
HEX Hexadecimal format.

inds Specification of the sequential input data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for QSAMIN is provided, no input is required.
- If the data set is cataloged specify: *dsname*[(*member*)],.
- If the data set is not cataloged specify: *dsname*[(*member*)],*volser*[,*unif*].

VSE:

- If a DLBL statement for SDSKIN is provided, no input is required.
- For VSAM-managed SAM specify: *dsname*,**YES**,*[ucat]*.
- For BAM-managed SAM specify: *dsname*,**NO**,*{unit|Vvolser}*.

CMS:

- Specify: *dsname*,*{unit|Vvolser}*.

| | |
|----------------|---|
| <i>member</i> | For a partitioned data set, the member name. |
| <i>mode</i> | Specify U to process physical blocks. The default is to process logical blocks. |
| <i>nlrecs</i> | Number of records to be printed or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are printed. |
| <i>recfmin</i> | Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: B Blocked F Fixed length U Undefined length V Variable length |
| <i>recsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>recsize</i> is required. |
| <i>skip</i> | Number of logical records to be skipped from the beginning of the data set. The default is 0. |
| <i>SYSnnn</i> | Programmer logical unit to which the input device is assigned. |
| <i>ucat</i> | For VSAM-managed SAM, the data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number for a non-cataloged data set. |
| <i>vsamin</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the input volume. |

Batch Example (MVS)

```

//SP JOB (acct),'name' Print QSAM Data
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//
//*
//STPSPEX EXEC DTOBAT
//SYSIN DD *
$$DITTO VER
$$DITTO SP DSNIN=SYS1.PROCLIB(COBUCLG)
$$DITTO EOJ
/*
  
```

Batch Example (VSE)

```

// JOB SP - Print SAM Data
// ASSGN SYS010,14D
// UPSI 1
// DLBL SDSKIN,'DTT.ITT.SAM.FB',0
// EXTENT ,SYSWK2
// EXEC DITTO
$$DITTO SP INPUT=SYS010,RECFMIN=FB,RECSIZE=80,
$$DITTO POSITION=5,NLRECS=1
$$DITTO EOJ
/&
  
```

SPB (Spool Browse)—VSE and CMS Only

Purpose Browse a VSE/POWER list queue or CP spool (reader queue) file. This function is available only in full-screen mode.

For more information about the SPB function, use the online help or see “Browsing Data” on page 32.

SPL (Spool List Display)—VSE Only

Purpose Displays a list of print (LST) entries in the list queue. This function is only available in full-screen mode.

For more information about the SPL function, use the online help.

SPS (Spool Services)—VSE Only

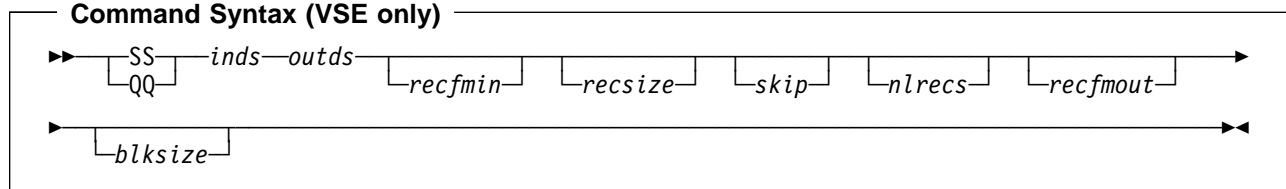
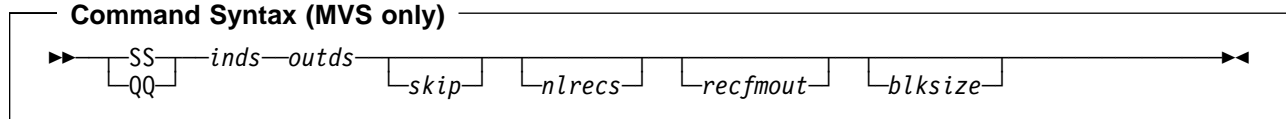
Purpose Displays all entries on a selected VSE POWER spool queue (LST, PUN, RDR, or XMT), and allows you to enter a command directly on an entry to browse, alter, delete, hold, release, or transfer it. This function is only available in full-screen mode.

For more information about the SPS function, use the online help.

SS, QQ (Sequential Data to Sequential Data)—MVS and VSE Only

Purpose Copy records from one sequential data set to another sequential data set.

For compatibility with previous releases, you can specify either SS or QQ.



Batch Syntax
 \$\$DITTO SS|QQ ...

Keyword Syntax
 DITTO \$\$\$|\$QQ ...

| MVS keywords | VSE keywords | CMS keywords |
|--|--|--------------|
| [{ INPUT={QSAMIN ddname} DSNIN=dsname [(member)] [VOLSERIN=volser] [UNITIN=unit] }] [POSITION={@ skip}] [NLRECS={ALL nlrecs}] [{ OUTPUT={QSAMOUT ddname} DSNOUT=dsname [(member)] [VOLSEROUT=volser] [UNITOUT=unit] [DISP={NEW OLD MOD CAT}] [SPACE=space] }] [RECFMOUT={U UA UM F FA FM FB FBA FBM FBS FBSA FBSM V VA VM VB VBA VBM VS VSA VSM VBS VBSA VBSM D DA DB DBA DS DSA DBS DBSA}] [BLKSIZE=blksize] | [INPUT={unit SYSnnn Vvolser}] [{ FILEIN={SDSKIN dlbl} DSNIN=dsname [VSAMIN={NO YES}] [UCATIN=ucat] }] [RECFMIN={U V VB F FB}] [RECSIZE=recsize] [POSITION={@ skip}] [NLRECS={ALL nlrecs}] [OUTPUT={unit SYSnnn Vvolser}] [{ FILEOUT={SDSKOUT dlbl} DSNOUT=dsname [VSAMOUT={NO YES}] [BEGIN=begin] [END=end] [UCATOUT=ucat] [DISP={NEW OLD}] [SPACE=space] [AVGSIZE=avgsiz] }] [RECFMOUT={U F V FB VB}] [BLKSIZE={MAX blksize}] | |

Parameters

avgsiz The average record size for a new VSAM-managed SAM data set.

begin The begin position for a BAM-managed SAM data set in one of the following forms:

cylhd Where the last two digits represent the head, and the other digits represent the cylinder (CKD)

Rnnnnnn To indicate the relative track number (CKD)

nnnnnn To indicate the physical block number (FBA)

blksize MVS: maximum length of an output block in bytes. Specify it to change the block size of an old data set, or to override any system default for a new data set.

VSE: the actual block size for record format FB, otherwise the maximum block size. For

record format FB or VB a block size is required, otherwise it is optional. Specify MAX to use the maximum block size allowed for the output device.

| | |
|---------------|---|
| <i>ddname</i> | Refers to a DD or TSO ALLOC statement. The default is QSAMIN for input, and QSAMOUT for output. |
| <i>disp</i> | Disposition of the output data set. MVS: specify OLD, MOD, NEW, or CAT for NEW,CATLG. The default is NEW. VSE: for VSAM-managed SAM, specify OLD or NEW. The default is NEW. |
| <i>dlbl</i> | Refers to a DLBL statement. The default is SDSKIN for input, and SDSKOUT for output. |
| <i>dsname</i> | Name of a sequential data set. If specified, any MVS DD or VSE DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table. |
| <i>end</i> | The end position for a BAM-managed SAM data set in one of the following forms: <i>cylhd</i> Where the last two digits represent the head, and the other digits represent the cylinder (CKD) <i>Rnnnnnn</i> To indicate the relative track number (CKD) <i>nnnnnnn</i> To indicate the physical block number (FBA) <i>Nnnnnnnn</i> To indicate the number of tracks (CKD) or blocks (FBA) to be allocated |
| <i>inds</i> | Specification of the sequential input data set. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If a DD or TSO ALLOC statement for QSAMIN is provided, no input is required. • If the data set is cataloged specify: <i>dsname[(member)],</i>. • If the data set is not cataloged specify: <i>dsname[(member)],volser[,unit].</i> VSE: <ul style="list-style-type: none"> • If a DLBL statement for SDSKIN is provided, no input is required. • For VSAM-managed SAM specify: <i>dsname,YES,[uca].</i> • For BAM-managed SAM specify: <i>dsname,NO,{unit volser}.</i> |
| <i>member</i> | For a partitioned data set, the member name. |
| <i>nlnrcs</i> | Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>outds</i> | Specification of the output data set. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If a JCL <i>ddname</i> or TSO ALLOC statement for QSAMOUT is provided, no input is required. • If the data set is cataloged, specify: <i>dsname[(member)],,{OLD MOD}</i> • If the data set is not cataloged, specify: <i>dsname[(member)],[,unit],{OLD MOD}</i> • For a new data set, specify: <i>dsname,volser[,unit],{NEW CAT MOD},space</i> SMS may modify the allocation of new data sets on your system. See your SMS Administrator for additional information. VSE: <ul style="list-style-type: none"> • If a DLBL statement for SDSKOUT is provided, no input is required. • For an existing VSAM-managed SAM, specify: <i>dsname,YES,[uca],OLD</i> • For a new VSAM-managed SAM, specify: <i>dsname,YES,[uca],NEW,space,avgsz</i> • For a BAM-managed SAM, specify: <i>dsname,NO,[unit volser],begin,end</i> |

| | |
|-----------------|---|
| <i>recfmin</i> | Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: B Blocked F Fixed length U Undefined length V Variable length |
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: U Undefined length F Fixed length V Variable length D Variable-length ISO/ANSI tape records B Blocked S Standard format (fixed length) or spanned format (variable length) A Containing ISO/ANSI control characters M Containing machine code control characters |
| | MVS: use <i>recfmout</i> to override existing DCB information. You cannot create a PDS with a |
| | record format of FBS. If output data sets with a record format containing the letter D are |
| | written to disk, unpredictable results may occur. |
| <i>recsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>recsize</i> is required. |
| <i>skip</i> | Number of logical records to be skipped from the beginning of the data set. The default is 0. |
| <i>space</i> | The number of space units to be allocated for a new data set. Specify <i>n</i> or (<i>n,m</i>), where <i>n</i> is the number of primary units, and <i>m</i> is the number of secondary units. MVS: The space unit is a track. VSE: The space unit is the length of an average record. |
| <i>SYSnnn</i> | Programmer logical unit to which the input or output device is assigned. |
| <i>ucat</i> | For VSAM-managed SAM, the data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE: Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number for a new or non-cataloged data set. |
| <i>vsamin</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>vsamout</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the input or output volume. |

Batch Example (MVS)

```

//SS JOB (acct),'name'
//* Copy sequential data to sequential data
/*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
/*
//STPSSEX EXEC DTOBAT
//SYS1INP DD DISP=SHR,
// DSN=SYS1.PARMLIB(LNKLST00)
//SYSIN DD *
$$DITTO VER
$$DITTO SS INPUT=SYS1INP,
$$DITTO DSNOUT=SIROED.TEMP.LINKLIST,
$$DITTO DISP=NEW,SPACE=(3,1),RECFMOUT=FB,
$$DITTO BLKSIZE=9040
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB SS - SAM to SAM ESDS with Reblocking
// UPSI 1
// EXEC DITTO
$$DITTO SS DSNIN=MYDATA.DISK,INPUT=244,
$$DITTO RECFMIN=FB,RECSIZE=255,
$$DITTO DSNOUT=MYDATA.NEW,VSAMOUT=YES,
$$DITTO UCATOUT=DTT.ITT.UCAT2,
$$DITTO OUTPUT=245,DISP=NEW,
$$DITTO SPACE=(500,50),AVGSIIZE=255,
$$DITTO RECFMOUT=FB,BLKSIZE=MAX
$$DITTO EOJ
/&

```

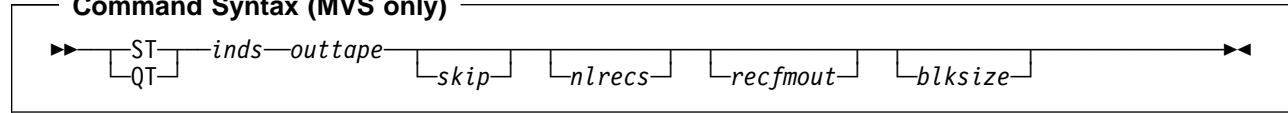
ST, QT (Sequential Data to Tape)

Purpose Copy records from a sequential data set to a tape file.

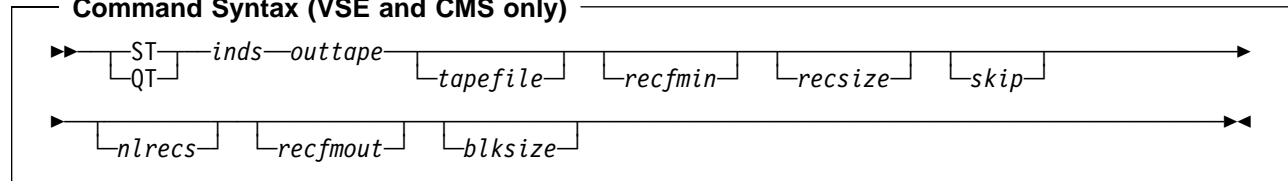
For compatibility with previous releases, you can specify either ST or QT.

CMS: CMS/DOS is required.

Command Syntax (MVS only)



Command Syntax (VSE and CMS only)



Batch Syntax

\$\$DITTO ST|QT ...

Keyword Syntax

DITTO \$ST|\$QT ...

| MVS keywords | VSE keywords | CMS keywords |
|--|--|--|
| [{ INPUT={QSAMIN ddname} DSNIN=dsname[(member)] [VOLSERIN=volser] [UNITIN=unit] }] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [LABEL=BLP] OUTPUT=ddname [DENS=mm] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] | [INPUT={unit SYSnnn Vvolser}] [{ FILEIN={SDSKIN dlbl} DSNIN=dsname [VSAMIN={NO YES}] [UCATIN=ucat] }] [RECFMIN={U V VB F FB}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] OUTPUT={unit SYSnnn} [DENS=mm] [{ FILEOUT={TAPEOUT tlbl} OUTFILE=tapefile }] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] | DSNIN=dsname INPUT={unit Vvolser} [RECFMIN={U V VB F FB}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] OUTPUT=unit [DENS=mm] [OUTFILE=tapefile] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] |

Parameters

blksize If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.

ddname Refers to a DD or TSO ALLOC statement. The default is QSAMIN for input. To create the output tape file with standard labels, specify AL or SL processing on the DD or TSO ALLOC statement.

dlbl Refers to a DLBL statement. The default is SDSKIN.

dsname Name of a sequential data set. If specified, any MVS DD or VSE DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table.

inds Specification of the sequential input data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for QSAMIN is provided, no input is required.
- If the data set is cataloged specify: *dsname*[(*member*)].
- If the data set is not cataloged specify: *dsname*[(*member*)],*volser*[,*unit*].

VSE:

- If a DLBL statement for SDSKIN is provided, no input is required.
- For VSAM-managed SAM specify: *dsname*,**YES**,[*uca*].
- For BAM-managed SAM specify: *dsname*,**NO**,{*unit*|**V***volser*}.

CMS:

- Specify: *dsname*,{*unit*|**V***volser*}.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
| the first DITTO function that uses the tape. For BLP processing requirements, see
| "Customizing the Security Environment" in the *DITTO/ESA Installation and Customization*
| *Guide*.

member For a partitioned data set, the member name.

mm A 2-byte tape mode code, as shown in "Tape Density and Mode Values" on page 67.

nlrecs Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.

outtape Specification of the output tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*[-*mm*].
- If label processing is desired specify:
ddname[-*mm*],[*tapevol*],*unit*,{**SL**|**AL**},*tapedsn*,[*seqnum*],[*xdate*].
- For NL or BLP tape processing specify:
ddname[-*mm*],[*tapevol*],*unit*,{**NL**|**BLP**},[*seqnum*],[*xdate*].

VSE or CMS:

- Specify: *unit*[-*mm*].

recfmin Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

B Blocked
F Fixed length
U Undefined length
V Variable length

recfmout Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

U Undefined length
F Fixed length
V Variable length
D Variable-length ISO/ANSI tape records
B Blocked
S Spanned format

recsize Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

| | |
|-----------------|--|
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>skip</i> | Number of logical records to be skipped from the beginning of the data set. The default is 0. |
| <i>SYSnnn</i> | Programmer logical unit to which the input or output device is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapefile</i> | VSE: the file ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed. CMS: the file ID of a tape file, if you want to perform label processing. By default, NL processing is performed. |
| <i>tapevol</i> | Serial number (external label or V0L1) of the tape volume to be mounted, or blank for a non-specific volume request. For a volume set, specify up to five volsers in the form (volser1, volser2...). |
| <i>tlbl</i> | Refers to a TLBL statement to process a labeled tape. The default is TAPEOUT. |
| <i>ucat</i> | For VSAM-managed SAM, the data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number for a non-cataloged data set. |
| <i>vsamin</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the input volume. |
| <i>xdate</i> | Expiration date in the form <code>[yy]yy.ddd</code> . |

Batch Example (MVS)

```

//ST JOB (acct),'name'
//* Copy Sequential file to SL Tape
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DELETE EXEC PGM=IEFBR14
//BSA1 DD DISP=(MOD,DELETE),UNIT=SYSALLDA,
// SPACE=(TRK,(1,1),RLSE),
// DSN=KEISTEW.BS43FB
//*
//STP1001 EXEC DTOBAT
//QSAMOUT DD DISP=(,CATLG),UNIT=SYSALLDA,
// SPACE=(CYL,(1,1),RLSE),
// RECFM=FBS,DSN=KEISTEW.BS43FB
//OUTTAPE DD DSN=TDALAIS.TAPEOUT,UNIT=CART,
// DISP=(,KEEP),LABEL=(2,SL),
// VOL=(,RETAIN,,,SER=DIT001)
//SYSIN DD *
$$DITTO VER
$$DITTO BS OUTPUT=QSAMOUT,NLRECS=5677,
$$DITTO RECSIZE=800,KEYLOC=1,KEYLEN=5,
$$DITTO INCR=5,RECFMOUT=FB,FILLCHAR=AN
$$DITTO ST INPUT=QSAMOUT,OUTPUT=OUTTAPE,
$$DITTO RECFMOUT=VB,BLKSIZE=6000
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB ST - SAM to NL Tape With Blocking
// UPSI 1
// DLBL INFILE,'DTT.ITT.SAM.U'
// ASSGN SYS010,DISK,VOL=SYSWK2,SHR
// EXEC DITTO
$$DITTO REW OUTPUT=A62 goto begin of tape
$$DITTO ST INPUT=SYS010,FILEIN=INFILE,
$$DITTO OUTPUT=A62,OUTFILE=TAPE.FILE,
$$DITTO RECFMOUT=VB,BLKSIZE=6000
$$DITTO EOJ
/&

```

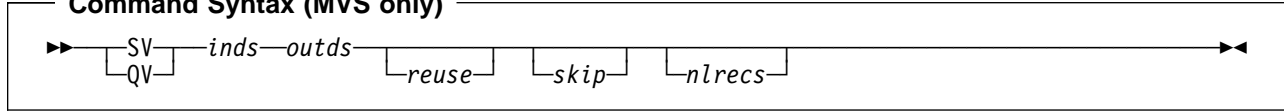
SV, QV (Sequential Data to VSAM)

Purpose Copy records from a sequential data set to a VSAM data set.

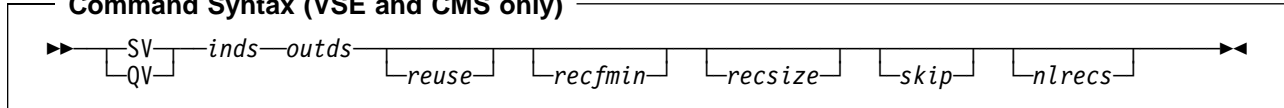
For compatibility with previous releases, you can specify either SV or QV.

CMS: CMS/VSAM is required.

Command Syntax (MVS only)



Command Syntax (VSE and CMS only)



Batch Syntax

\$\$\$DITTO SV|QV ...

Keyword Syntax

DITTO \$\$SV|\$QV ...

| MVS keywords | VSE keywords | CMS keywords |
|--|---|---|
| <pre> [{ INPUT={QSAMIN ddname} DSNIN=dsname[(member)] [VOLSERIN=volser] [UNITIN=unit] }] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [{ OUTPUT={VSAMOUT ddname} DSNOUT=dsname }] [REUSE={NO YES}] </pre> | <pre> [INPUT={unit SYSnnn Vvolser}] [{ FILEIN={SDSKIN dlbl} DSNIN=dsname [VSAMIN={NO YES}] [UCATIN=ucatl] }] [RECFMIN={U V VB F FB}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [{ FILEOUT={VDSKOUT dlbl} DSNOUT=dsname [UCATOUT=ucatl] }] [REUSE={NO YES}] [PWDOUT=pwdout] </pre> | <pre> DSNIN=dsname INPUT={unit Vvolser} [RECFMIN={U V VB F FB}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] DSNOUT=dsname [UCATOUT=ucatl] [REUSE={NO YES}] [PWDOUT=pwdout] </pre> |

Parameters

ddname Refers to a DD or TSO ALLOC statement. The default is QSAMIN for input, and VSAMOUT for output.

dlbl Refers to a DLBL statement. The default is SDSKIN for input, and VDSKOUT for output.

dsname Name of a sequential data set or VSAM data set.

If you specify a sequential data set name, any MVS DD or VSE DLBL statements provided are not used. To further describe the sequential data set, use the appropriate keywords as shown in the table.

inds Specification of the sequential input data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for QSAMIN is provided, no input is required.
- If the data set is cataloged specify: *dsname*[(*member*)],.
- If the data set is not cataloged specify: *dsname*[(*member*)],*volser*[,*unif*].

VSE:

- If a DLBL statement for SDSKIN is provided, no input is required.
- For VSAM-managed SAM specify: *dsname*,**YES**,*[ucat]*.
- For BAM-managed SAM specify: *dsname*,**NO**,*{unit|Vvolser}*.

CMS:

- Specify: *dsname*,*{unit|Vvolser}*.

| | |
|---------------|--|
| <i>member</i> | For a partitioned data set, the member name. |
| <i>nlrecs</i> | Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>outds</i> | Specification of the VSAM output data set. (Command Syntax only.) |

MVS:

- If a DD or TSO ALLOC statement for VSAMOUT is provided, no input is required.
- If no DD statement is provided specify: *dsname*.

VSE:

- If a DLBL statement for VDSKOUT is provided, no input is required.
- If no DLBL statement is provided specify: *dsname*,*[ucat]*.

CMS:

- Specify: *dsname*,*[ucat]*.

| | |
|----------------|---|
| <i>pwdout</i> | Password of a password-protected VSAM data set. |
| <i>recfmin</i> | Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: B Blocked F Fixed length U Undefined length V Variable length |
| <i>recsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>recsize</i> is required. |
| <i>reuse</i> | Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO. |
| <i>skip</i> | Number of logical records to be skipped from the beginning of the data set. The default is 0. |
| <i>SYSnnn</i> | Programmer logical unit to which the input device is assigned. |
| <i>ucat</i> | For VSAM or VSAM-managed SAM, the data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number for a non-cataloged data set. |
| <i>vsamin</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the input volume. |

Batch Example (MVS)

```

//SV JOB (acct),'name' QSAM to VSAM
//*
//IDCPROC PROC
//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DMBSV11 EXEC IDCPROC,REGION=6M
//SYSIN DD *
DELETE KEISTEW.TEMP.VSAM.OSV1107 CLUSTER -
PURGE ERASE
DEFINE CLUSTER ( -
NAME( KEISTEW.TEMP.VSAM.OSV1107 ) -
CISZ(4096) -
NUMBERED -
RECORDS(1000 100) -
RECORDSIZE(57 57) -
REUSE -
VOL(MVS4W1) ) -
DATA( -
NAME(KEISTEW.TEMP.VSAM.OSV1107.DATA) ) -
/*
//DMBS111 EXEC DTOBAT,REGION=6M
//OBS07 DD DISP=(,CATLG),UNIT=SYSDA,
// SPACE=(TRK,(3,12,20),RLSE),
// RECFM=FB,LRECL=57,
// DSN=KEISTEW.TEMP.ISV1107
//SYSIN DD *
$$DITTO BS DSNOUT=KEISTEW.TEMP.ISV1107(MEM1),
$$DITTO RECFMOUT=FB,FILLCHAR=AN,
$$DITTO NLRECS=1087,RECSIZE=57,
$$DITTO BLKSIZE=11400,DISP=OLD
/*
//STP1007 EXEC DTOBAT
//SYSIN DD *
$$DITTO SV DSNIN=KEISTEW.TEMP.ISV1107(MEM1),
$$DITTO DSNOUT=KEISTEW.TEMP.VSAM.OSV1107,
$$DITTO NLRECS=167
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB SV - SAM to VSAM
// UPSI 1
// DLBL IJSYSCT,'DTT.ITT.UCAT2',,VSAM
// DLBL SDSKIN,'DTT.ITT.SAMDS.FB',,VSAM
// DLBL VDSKOUT,'WORK.ESDS',,VSAM
// EXEC DITTO
$$DITTO SV RECSIZE=230,REUSE=YES
$$DITTO EOJ
/&

```

TB (Tape Browse)

Purpose Browse tape records at the terminal. This function is available only in full-screen mode.

For more information about the TB function, use the online help or see "Browsing Data" on page 32.

TC (Tape to Card)

Purpose Transfer card-image records from a tape to cards. Output is written to:

- MVS: SYSPUNCH
- VSE: SYSPCH
- CMS: The virtual punch address 00D.

The input records must have a record length of 80, 81 (VSE only), or a multiple of 80.

Command Syntax

```

▶▶ TC intape [recfmin] [interpret]

```

Batch Syntax

```

$$DITTO TC ...

```

Keyword Syntax

```

DITTO $TC ...

```

| MVS keywords | VSE keywords | CMS keywords |
|---|---|--|
| [LABEL=BLP] INPUT=ddname [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [INTERPRET={NO YES}] | INPUT={unit SYSnnn} [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [INTERPRET={NO YES}] | INPUT=unit [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [INTERPRET={NO YES}] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

interpret Interpret the output. If you specify YES, the output device must be a card punch with the card print feature installed. The default is NO.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
 | the first DITTO function that uses the tape. For BLP processing requirements, see
 | "Customizing the Security Environment" in the *DITTO/ESA Installation and Customization*
 | *Guide*.

mm A 2-byte tape mode code, as shown in "Tape Density and Mode Values" on page 67.

recfmin Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

- B** Blocked
- D** Variable-length ISO/ANSI tape records
- F** Fixed length
- S** Spanned format

| | | |
|----------------|----------|--|
| | U | Undefined length |
| | V | Variable length |
| <i>seqnum</i> | | Relative position of the data set on the tape. |
| <i>SYSnnn</i> | | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapevol</i> | | Serial number (external label or V0L1) of the tape volume to be mounted. |
| <i>unit</i> | | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | | Expiration date in the form <i>[yy]yy.ddd</i> . |

TCN (Tape to Console)

Purpose Display all or part of a tape record at the console, in character or hexadecimal format. The tape record or tape mark following the current tape position is displayed. Use SET ASCII=IN to display data stored in ASCII format on tape.

This function is available only in line mode. If you enter TCN in full-screen mode, the TB function is invoked. For more information about this function, see “Viewing a Physical Tape Record at the System Console” on page 77.

Command Syntax

▶▶—TCN—*intape*————▶▶

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inoutf].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inoutf].

VSE or CMS:

- Specify: *unit[-mm]*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

seqnum Relative position of the data set on the tape.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or VOL1) of the tape volume to be mounted.

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form [*yy*]*yy.ddd*.

Subcommands

D,*n* Displays data in character format, starting with byte *n* if specified.

DX,*n* Displays data in hexadecimal updown format, starting with byte *n* if specified.

P Prints the current record in character format.

PX Prints the current record in hexadecimal updown format.

B Moves back to the previous record.

N Moves forward to the next record.

? Gives help information.

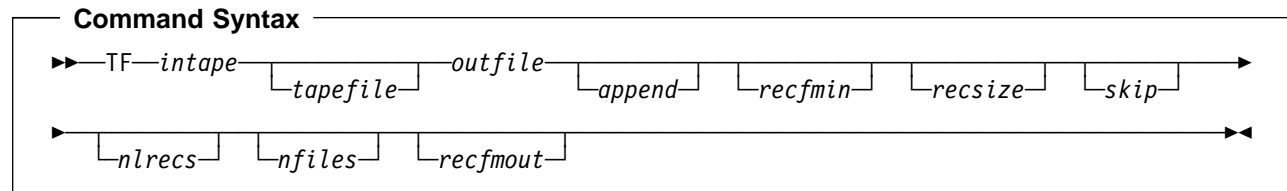
QUIT Ends the TCN function. The tape is positioned after the current record.

You can also use DITTO tape-positioning commands (BSF, FSF, BSR, FSR, and REW) within the TCN function.

TF (Tape to CMS File)—CMS Only

Purpose Copy tape records to a CMS file.

This function can be used to prepare tape data for distribution across networks. For more information, see “Copying Tape Data across a Network” on page 81.



| Batch Syntax | Keyword Syntax |
|------------------|----------------|
| \$\$DITTO TF ... | DITTO \$TF ... |

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--------------|---|
| | | INPUT = <i>unit</i> [INFILE] = <i>tapefile</i> [RECFMIN] = {U F FB V VB VBS VS D DB DBS DS} [RECSIZE] = <i>recsize</i> [POSITION] = {0 skip} [NLRECS] = {ALL nlrecs} [NFILES] = {1 EOV EOD nfiles} OUTFILE = <i>'outfile'</i> [APPEND] = {NO YES} [RECFMOUT] = {V U F} |

Parameters

append Specify YES to add output to the end of an existing CMS file. By default, any existing CMS file with the specified file ID is replaced.

intape Specification of the input tape unit. (Command Syntax only.) Specify: *unit[-mm]*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

nfiles Number of tape files to be copied. The maximum is 99999; the default is 1.

EOV Copies files until a double tape mark or an EOV label is reached.

EOD Copies files until the delimiter specified in the EOD parameter of the SET function is reached.

Use this parameter to copy a multifile tape, including tape marks, to a single CMS file. (You can use the FT function to copy the CMS file back to tape, including the tape marks). For more information, see “Copying Tape Data across a Network” on page 81.

For labeled tape processing, only one file is processed. Specifying *nfiles* is ignored.

nlrecs Number of records to be copied or ALL. The maximum number is 99999999. If you specify ALL or omit the parameter, all the remaining records are copied.

outfile Specifies the CMS output file identifier (Command Syntax only): *filename filetype [filemode]*, enclosed in single quotes. The default filemode is A.

recfmin Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

B Blocked

TF

- D** Variable-length ISO/ANSI tape records
- F** Fixed length
- S** Spanned format
- U** Undefined length
- V** Variable length

recfmout Record format for the output:

- F** Checks that all input records are fixed-length, and creates a fixed format output file.
- V** or **U** Processes the input records without a length check, and creates a variable-format output file (the default).

recsize Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

skip Number of logical records to be skipped. The default is 0.

tapefile File ID of a tape file, if you want to perform label processing. By default, NL processing is performed.

unit Device number (3 or 4 hexadecimal digits).

Keyword Example (CMS)

```
/* REXX - copy SL tape file to CMS file */
"DITTO $REW INPUT 181" /* goto begin of tape */
"DITTO $TF INPUT 181 INFILE TEST.DATA" ,
"OUTFILE 'TEST DATA A'"
Exit rc
```

TFA (Print SYSLST Tape Type A)—VSE Only

Purpose Print a SYSLST tape file that contains CCW command codes. To print a SYSLST tape file that contains ASA control characters, use TFD. (Files produced by VSE job control contain ASA control characters.)

For further information on the use of this function, see “Printing Tape Files” on page 79.

Command Syntax

```
▶▶ TFA intape [nlrecs] ◀◀
```

Batch Syntax

```
$$DITTO TFA ...
```

Keyword Syntax

```
DITTO $TFA ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | INPUT={unit SYSnnn} [NLRECS={ALL nlrecs}] | |

Parameters

- intape* Specification of the input tape unit. (Command Syntax only.) Specify: *unit*[*-mm*].
- mm* A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.
- nlrecs* Number of records to be printed or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are printed.
- SYSnnn* Programmer logical unit to which the tape is assigned.
- unit* Device number (3 or 4 hexadecimal digits).

Batch Example (VSE)

```
// JOB TFA - Print SYSLST Tape Type A
// UPSI 1
// EXEC DITTO
$$DITTO REW INPUT=A61 go to begin of tape
$$DITTO TFA INPUT=A61
$$DITTO EOJ
/&
```

TFD (Print SYSLST Tape Type D)—VSE Only

Purpose Print a SYSLST tape file that contains ASA control characters. To print a SYSLST tape file that contains CCW command codes, use TFA. (Files produced by VSE job control contain ASA control characters.)

For further information on the use of this function, see “Printing Tape Files” on page 79.

Command Syntax

```
▶—TFD—intape [ nlrecs ]◀
```

Batch Syntax

```
$$DITTO TFD ...
```

Keyword Syntax

```
DITTO $TFD ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | INPUT={ <i>unit</i> SYS <i>nnn</i> } [NLRECS={ALL <i>nlrecs</i> }] | |

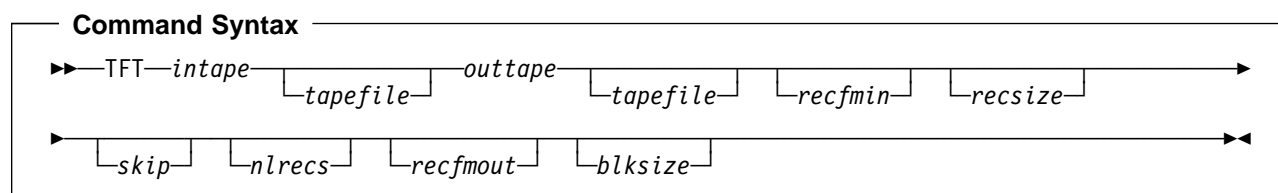
Parameters

- intape* Specification of the input tape unit. (Command Syntax only.) Specify: *unit*[-*mm*].
- mm* A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.
- nlrecs* Number of records to be printed or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are printed.
- SYS*nnn* Programmer logical unit to which the tape is assigned.
- unit* Device number (3 or 4 hexadecimal digits).

TFT (Tape File to Tape File)—VSE and CMS Only

Purpose Copy one file, or part of one file, from a standard labeled tape or an unlabeled tape to a standard labeled tape. You can change the record format and reblock the tape file.

For further information about the use of this function, see “Copying Tape Records” on page 80. For further information about other tape copy functions, see “Copying from Tape to Tape” on page 80.



Batch Syntax

```

$$$DITTO TFT ...
  
```

Keyword Syntax

```

DITTO $TFT ...
  
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--|
| | INPUT ={unit SYSnnn} [{ FILEIN={TAPEIN tbl} INFILE=tapefile }] [RECFMIN ={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE =recsize] [POSITION ={0 skip}] [NLRECS ={ALL nlrecs}] OUTPUT ={unit SYSnnn} [DENS =mm] [{ FILEOUT={TAPEOUT tbl} OUTFILE=tapefile }] [RECFMOUT ={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE =blksize] | INPUT =unit [INFILE =tapefile] [RECFMIN ={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE =recsize] [POSITION ={0 skip}] [NLRECS ={ALL nlrecs}] OUTPUT =unit [DENS =mm] [OUTFILE =tapefile] [RECFMOUT ={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE =blksize] |

Parameters

- blksize** If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.
- intape** Specification of the input tape unit. (Command Syntax only.) Specify: *unit*[-*mm*].
- mm** A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.
- nlrecs** Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.
- outtape** Specification of the output tape unit. (Command Syntax only.) Specify: *unit*[-*mm*].
- recfmin** Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:
- B** Blocked
 - D** Variable-length ISO/ANSI tape records
 - F** Fixed length
 - S** Spanned format
 - U** Undefined length

TFT

V Variable length

recfmout Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

U Undefined length

F Fixed length

V Variable length

D Variable-length ISO/ANSI tape records

B Blocked

S Spanned format

recsize Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

skip Number of logical records to be skipped. The default is 0.

SYSnnn Programmer logical unit to which the tape is assigned.

tapefile VSE: the file ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed.

CMS: the file ID of a tape file, if you want to perform label processing. By default, NL processing is performed.

tlbl Refers to a TLBL statement to process a labeled tape. The default is TAPEIN for input, and TAPEOUT for output.

unit Device number (3 or 4 hexadecimal digits).

Batch Example (VSE)

```
// JOB TFT - Copy SL tape file to SL tape file
// UPSI 1
// EXEC DITTO
$$DITTO REW INPUT=A61      goto begin of tape
$$DITTO REW OUTPUT=A62    goto begin of tape
$$DITTO TFT INPUT=A61,INFILE=TAPE.DATA,
$$DITTO      OUTPUT=A62,OUTFILE=TAPE.COPY,
$$DITTO      POSITION=5,NLRECS=10
$$DITTO EOJ
/&
```

Keyword Example (CMS)

```
/* REXX - copy SL tape file to NL tape file */
"DITTO $REW INPUT 181" /* goto begin of tape */
"DITTO $REW OUTPUT 182" /* goto begin of tape */
"DITTO $TFT INPUT 181 INFILE TAPE.DATA" ,
      "OUTPUT 182"
Exit rc
```

TL (Tape to Library Member)—VSE Only

Purpose Copy a tape file to a library member.

Command Syntax

```
▶—TL—intape—tapefile—recfmin—outlib—member—recfmout—▶
```

Batch Syntax

```
$$DITTO TL ...
```

Keyword Syntax

```
DITTO $TL ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | INPUT ={ <i>unit</i> SYS <i>nnn</i> } [{ FILEIN ={ TAPEIN <i>tlbl</i> } INFILE = <i>tapefile</i> }] [RECFMIN ={ U F FB V VB VBS VS D DB DBS DS }] LIBOUT = <i>lib</i> MEMBEROUT = <i>member</i> [RECFMOUT ={ F S }] | |

Parameters

- intape* Specification of the input tape unit. (Command Syntax only.) Specify: *unit*[-*mm*].
- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- member* Name of the library member in the form *name.type*.
- mm* A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.
- outlib* Specification of the output library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucaf]*.
- recfmin* Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:
- B** Blocked
 - D** Variable-length ISO/ANSI tape records
 - F** Fixed length
 - S** Spanned format
 - U** Undefined length
 - V** Variable length
- recfmout* Record format for the output:
- F** Fixed-length, 80 (the default)
 - S** String format (always used if the member type is PHASE).
- SYS*nnn*** Programmer logical unit to which the tape is assigned.
- tapefile* File ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed.
- tlbl* Refers to a TLBL statement to process a labeled tape. The default is TAPEIN.

TL

ucat The data set name of the user catalog.
unit Device number (3 or 4 hexadecimal digits).
volser Volume serial number of the device where the library is located.

Batch Example (VSE)

```
// JOB TL - NL Tape to Library Member  
// UPSI 1  
// EXEC DITTO  
$$DITTO TL INPUT=A61,LIBOUT=PRD1.BASE,  
$$DITTO    MEMBEROUT=DITTO.PHASE  
$$DITTO EOJ  
/&
```


TLB (Tape Label Display)

Purpose Print header and trailer label records for all the data sets on the tape. A summary of the label data is printed at the end of the listing. You can also choose a short listing, which prints only the VOL1 label and the first header label set.

For further information on the use of this function, see “Finding Out What a Tape Contains” on page 71.

Note: Under MVS, if you do not know the volser and first data set name of the tape, specify BLP during allocation. If your installation does not allow BLP usage but DITTO is running APF-authorized, DITTO forces BLP for this function; for security reasons, the tape is freed (unallocated) when TLB ends. In batch mode, you must code a deferred mount in the DD statement.

| Command Syntax | | |
|--|--|---|
| | | |
| Batch Syntax | Keyword Syntax | |
| \$\$DITTO TLB ... | DITTO \$TLB ... | |
| MVS keywords | VSE keywords | CMS keywords |
| [LABEL=BLP] {INPUT=ddname OUTPUT=ddname} [TYPE={LONG SHORT}] | {INPUT={unit SYSnnn} OUTPUT={unit SYSnnn}} [TYPE={LONG SHORT}] | {INPUT=unit OUTPUT=unit} [TYPE={LONG SHORT}] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
| the first DITTO function that uses the tape. For BLP processing requirements, see
| “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization*
| *Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

seqnum Relative position of the data set on the tape.

SYSnnn Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

TLB

- tapevol* Serial number (external label or VOL1) of the tape volume to be mounted. For a volume set, specify up to five volsers in the form (volser1, volser2...).
- type* Specify one of the following:
LONG Print all header and trailer label records for all data sets on the tape (the default).
SHORT Print only the VOL1 label and the header labels for the first data set.
- unit* MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
VSE or CMS: Device number (3 or 4 hexadecimal digits).
- xdate* Expiration date in the form [yy]yy.ddd.

Batch Example (MVS)

```
//TLB JOB (acct),'name' Tape Label Display
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DITTO EXEC DTOBAT
//SYSPRINT DD SYSOUT=*
//TAPE DD UNIT=561,VOL=SER=DIT001,
// DISP=SHR,LABEL=(,BLP)
//SYSIN DD *
$$DITTO TLB INPUT=TAPE
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB TLB - Tape Label Display
// UPSI 1
// EXEC DITTO
$$DITTO TLB INPUT=A61,TYPE=SHORT
$$DITTO EOJ
/&
```

TLT (Tape to Labeled Tape)

Purpose Copy standard labeled tapes with label processing. Copying multivolume and multifile tapes is supported.

For further information on the use of this function, see “Copying Tape Records” on page 80.

Command Syntax (MVS only)

```

  >> TLT intape outtape [nfiles] [newvolser] [type]
  
```

Command Syntax (VSE and CMS only)

```

  >> TLT intape [tapefile] outtape [nfiles] [newvolser] [type]
  
```

Batch Syntax

```

  $$DITTO TLT ...
  
```

Keyword Syntax

```

  DITTO $TLT ...
  
```

| MVS keywords | VSE keywords | CMS keywords |
|---|---|---|
| [LABEL=BLP] INPUT=ddname OUTPUT=ddname [DENS=mm] [NFILES={ALL nfiles}] [NEWVOL={* newvolser}] [TYPE={SL MIXED}] | INPUT={unit SYSnnn} OUTPUT={unit SYSnnn} [{ FILEIN={TAPEIN tlbl} INFILE=tapefile }] [DENS=mm] [NFILES={ALL nfiles}] [NEWVOL={* newvolser}] [TYPE={SL MIXED}] | INPUT=unit [INFILE=tapefile] OUTPUT=unit [DENS=mm] [NFILES={ALL nfiles}] [NEWVOL={* newvolser}] [TYPE={SL MIXED}] |

Parameters

- ddname* Refers to a DD or TSO ALLOC statement.
- inout* Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).
- intape* Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
 | the first DITTO function that uses the tape. For BLP processing requirements, see
 | “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization*
 | *Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

newvolser Serial number (VOL1) for the initialization of the output volumes. Specify up to five serial numbers in the form (volser1, volser2,...) or specify an asterisk (*) to copy the volume serial numbers of the input volumes. If omitted, the serial numbers of the output volumes are not changed.

TLT

| | |
|-----------------|---|
| <i>nfiles</i> | Number of tape files to be copied or ALL. The maximum is 99 999; the default is ALL. |
| <i>outtape</i> | Specification of the output tape unit. (Command Syntax only.) |
| | MVS: |
| | <ul style="list-style-type: none">• If the tape is already allocated specify: <i>ddname[-mm]</i>.• If label processing is desired specify: <i>ddname[-mm],[tapevol],unit,{SL AL},tapedsn,[seqnum],[xdate]</i>.• For NL or BLP tape processing specify: <i>ddname[-mm],[tapevol],unit,{NL BLP},[seqnum],[xdate]</i>. |
| | VSE or CMS: |
| | <ul style="list-style-type: none">• Specify: <i>unit[-mm]</i>. |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapefile</i> | The file ID of the first tape file for verification. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted, or blank for a non-specific volume request. For a volume set, specify up to five volsers in the form (volser1, volser2...). |
| <i>tlbl</i> | Refers to a TLBL statement for label verification. The default is TAPEIN. |
| <i>type</i> | Specify one of the following: MIXED When copying a tape containing both labeled and unlabeled files. SL When copying a standard labeled tape (the default). DITTO will indicate an error if the input tape does not have IBM standard labels. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form <i>[yy]yy.ddd</i> . Use this parameter only to identify a special tape for a tape management system. |

Batch Example (MVS)

```
//TLT JOB (acct),'name' Tape to Labeled Tape
//DITTO EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//TAPE1 DD UNIT=(381,,DEFER),LABEL=(,BLP),
// VOL=SER=(DITTO1,DITTO2),DISP=OLD
//TAPE2 DD UNIT=(382,,DEFER),LABEL=(,BLP),
// VOL=SER=(DITTO3,DITTO4),DISP=OLD
//SYSIN DD *
$$DITTO TLT INPUT=TAPE1,OUTPUT=TAPE2,NFILES=ALL
$$DITTO EOJ
//
```

Batch Example (VSE)

```
// JOB TLT - Labeled Tape To Labeled Tape
// UPSI 1
// EXEC DITTO
$$DITTO REW INPUT=A61 goto begin of tape
$$DITTO REW OUTPUT=A62 goto begin of tape
$$DITTO TLT INPUT=A61,OUTPUT=A62,NFILES=ALL
$$DITTO EOJ
/&
```

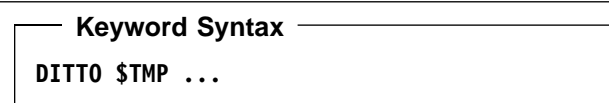
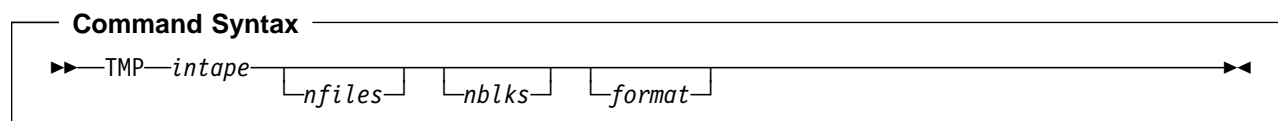
Note: Any alternate tape device assignment will be ignored by this function. DITTO prompts the operator for a continuation tape as required.

TMP (Tape Map)

Purpose Summarize tape contents. For a specified number of files on the tape, this function prints the contents of the first block or blocks, the number of records, the minimum and maximum block sizes, the size of the file, and the amount of tape used.

You can summarize the contents of the entire tape, or you can limit the summary to a specific number of tape files, and, within each tape file, to a specific number of blocks.

For further information about the use of this function, see “Finding Out What a Tape Contains” on page 71.



| MVS keywords | VSE keywords | CMS keywords |
|--|---|---|
| [LABEL=BLP] {INPUT=ddname OUTPUT=ddname} [NFILES={1 EOV EOD nfiles}] [NBLKS={1 nblks}] [FORMAT={HEX CHAR}] | {INPUT={unit SYSnnn} OUTPUT={unit SYSnnn}} [NFILES={1 EOV EOD nfiles}] [NBLKS={1 nblks}] [FORMAT={HEX CHAR}] | {INPUT=unit OUTPUT=unit} [NFILES={1 EOV EOD nfiles}] [NBLKS={1 nblks}] [FORMAT={HEX CHAR}] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

format The format of the output:

- HEX** Hexadecimal format (the default)
- CHAR** Character format.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
 | the first DITTO function that uses the tape. For BLP processing requirements, see
 | “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization*
 | *Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

nblks Number of tape blocks to be printed from each file. The range is 0 to 99 999 999; the default is 1.

TMP

- nfiles* Number of tape files to be summarized. The maximum is 99999; the default is 1.
- EOV** Summarizes files until a double tape mark or an EOV label is reached.
- EOD** Summarizes files until the delimiter specified in the EOD parameter of the SET function is reached. The EOD delimiter is detected only if it appears in the block or blocks specified by *nblks*.
- seqnum* Relative position of the data set on the tape.
- SYSnnn* Programmer logical unit to which the tape is assigned.
- tapedsn* Name of the tape data set that you want to work with, if label processing is used.
- tapevol* Serial number (external label or V0L1) of the tape volume to be mounted.
- unit* MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
VSE or CMS: Device number (3 or 4 hexadecimal digits).
- xdate* Expiration date in the form [yy]yy.ddd.

Batch Example (MVS)

```
//TMP JOB (acct),'name' Tape Map
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//
//*
//DITTO EXEC DTOBAT
//SYSPRINT DD SYSOUT=*
//TAPETMP DD UNIT=561,VOL=SER=DIT001,
// DISP=SHR,LABEL=(,BLP)
//SYSIN DD *
$$DITTO TMP INPUT=TAPETMP,NFILES=EOV
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB TMP - Tape Map
// UPSI 1
// EXEC DITTO
$$DITTO REW INPUT=580
$$DITTO TMP INPUT=580,FORMAT=CHAR,
$$DITTO NBLKS=5,NFILES=EOV
$$DITTO EOJ
/*
/&
```

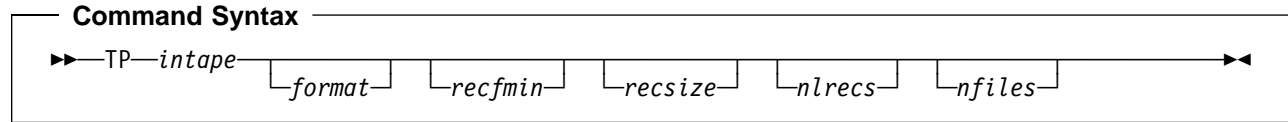
Keyword Example (CMS)

```
/* REXX - Tape Map */
"DITTO $REW INPUT 181" /* goto begin of tape */
"DITTO $TMP INPUT 181 NBLKS 0 NFILES EOV"
Exit rc
```

TP (Tape Print)

Purpose Print tape files in character or hexadecimal format. Printing starts with the record following the current position, and includes any leading tape mark. You can specify the number of files to be printed; otherwise printing continues up to (and including) the next tape mark.

For further information on the use of this function, see “Printing Tape Files” on page 79.



Batch Syntax

```

    $$$DITTO TP ...
  
```

Keyword Syntax

```

    DITTO $TP ...
  
```

| MVS keywords | VSE keywords | CMS keywords |
|--|--|---|
| [LABEL=BLP] {INPUT=ddname OUTPUT=ddname} [FORMAT={CHAR HEX}] [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [NLRECS={ALL nlrecs}] [NFILES={1 EOVS EOD nfiles}] | {INPUT={unit SYSnnn} OUTPUT={unit SYSnnn}} [FORMAT={CHAR HEX}] [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [NLRECS={ALL nlrecs}] [NFILES={1 EOVS EOD nfiles}] | {INPUT=unit OUTPUT=unit} [FORMAT={CHAR HEX}] [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [NLRECS={ALL nlrecs}] [NFILES={1 EOVS EOD nfiles}] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

format The format of the output:
CHAR Character format (the default)
HEX Hexadecimal format.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
 | the first DITTO function that uses the tape. For BLP processing requirements, see
 | “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization*
 | *Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

nfiles Number of tape files to be printed. The maximum is 99999; the default is 1.

EOV Prints files until a double tape mark or an EOVS label is reached.

EOD Prints files until the delimiter specified in the EOD parameter of the SET function is reached.

If the maximum number of records per file (*nlrecs*) is reached, printing stops and the function ends regardless of the *nfiles* value.

| | |
|----------------|---|
| <i>nlrecs</i> | Maximum number of records to be printed from any one file. If this number is exceeded, printing stops. This parameter provides protection against printing a file that is much larger than expected. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records in the file, up to and including the next tape mark, are printed. |
| <i>recfmin</i> | Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <ul style="list-style-type: none"> B Blocked D Variable-length ISO/ANSI tape records F Fixed length S Spanned format U Undefined length V Variable length |
| <i>recsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>recsize</i> is required. |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form [<i>yy</i>] <i>yy.ddd</i> . |

TRL (Tape Record Load)

Purpose Alter the contents or length of records within a tape file. The altered tape file is written to the output tape.

This function is not available in batch mode or command mode.

For further information on the use of this function, see “Updating One Physical Record at a Time” on page 75.

Command Syntax

```
▶▶—TRL—intape—outtape—◀◀
```

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

outtape Specification of the output tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname[-mm]*.
- If label processing is desired specify:
ddname[-mm],[tapevol],unit,{SL|AL},tapedsn,[seqnum],[xdate].
- For NL or BLP tape processing specify:
ddname[-mm],[tapevol],unit,{NL|BLP},[seqnum],[xdate].

VSE or CMS:

- Specify: *unit[-mm]*.

seqnum Relative position of the data set on the tape.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or VOL1) of the tape volume to be mounted, or blank for a non-specific volume request.

TRL

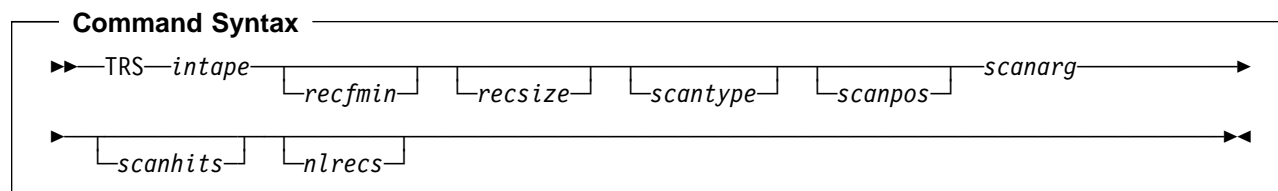
unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form *[yy]yy.ddd*.

TRS (Tape Record Scan)

Purpose Locate data within a tape file.

For further information on the use of this function, see “Searching for Specific Data” on page 78.



Batch Syntax

```

    $$DITTO TRS ...
  
```

Keyword Syntax

```

    DITTO $TRS ...
  
```

| MVS keywords | VSE keywords | CMS keywords |
|---|---|--|
| [LABEL=BLP] {INPUT=ddname OUTPUT=ddname} [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [SCANTYPE={D DP}] [SCANPOS={1 scanpos}] SCANARG=scanarg [SCANHITS={1 ALL scanhits}] [NLRECS={ALL nlrecs}] | {INPUT={unit SYSnnn} OUTPUT={unit SYSnnn}} [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [SCANTYPE={D DP}] [SCANPOS={1 scanpos}] SCANARG=scanarg [SCANHITS={1 ALL scanhits}] [NLRECS={ALL nlrecs}] | {INPUT=unit OUTPUT=unit} [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [SCANTYPE={D DP}] [SCANPOS={1 scanpos}] SCANARG=scanarg [SCANHITS={1 ALL scanhits}] [NLRECS={ALL nlrecs}] |

Parameters

- ddname* Refers to a DD or TSO ALLOC statement.
- inout* Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).
- intape* Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

nlrecs Number of records to be scanned or ALL. The maximum number is 99999999. If you specify ALL or omit the parameter, DITTO scans to the end of the file or until the specified number of hits is found.

TRS

| | |
|-----------------|--|
| <i>recfmin</i> | Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: B Blocked D Variable-length ISO/ANSI tape records F Fixed length S Spanned format U Undefined length V Variable length |
| <i>recsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>recsize</i> is required. |
| <i>scanarg</i> | The string to be searched for. The string can contain up to 50 characters and can be: <ul style="list-style-type: none">• A character string. The string cannot contain blanks or commas. Case (uppercase or lowercase) is ignored.• A character string enclosed in single quotation marks. The string <i>can</i> contain blanks and commas; the string must match exactly (including case).• X followed by a hexadecimal string enclosed in single quotation marks, for example, X'C1C2C3'. |
| <i>scanhits</i> | Specifies whether scanning stops after the first match is found, after a specified number of matches, or at the end of the file. The maximum is 99 999 999; the default is 1. Specify ALL to scan to the end of the file. |
| <i>scanpos</i> | Offset (byte position) where the search begins within a record. The default is 1. |
| <i>scantype</i> | One of the following: D Scans records <i>only</i> at the position in the record specified by <i>scanpos</i> (the default) DP Scans records <i>starting</i> at the position specified by <i>scanpos</i> |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapevol</i> | Serial number (external label or V0L1) of the tape volume to be mounted. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form [yy]yy.ddd. |

Batch Example (MVS)

```

| //TRS JOB (acct),'name' Tape Record Scan
| //*
| //DTOBAT PROC
| //DTOBAT EXEC PGM=DITTO
| //STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
| //SYSPRINT DD SYSOUT=*
| //SYSABEND DD SYSOUT=*
| //SYSUDUMP DD SYSOUT=*
| // PEND
| //*
| //DITTO EXEC DTOBAT
| //SYSPRINT DD SYSOUT=*
| //TAPE DD UNIT=561,VOL=SER=DIT001,
| // DISP=SHR,LABEL=(,BLP)
| //SYSIN DD *
| $$DITTO TRS INPUT=TAPE,SCANARG='HDR2'
| $$DITTO EOJ
| /*

```

Batch Example (VSE)

```

| // JOB TRS - Tape Record Scan
| // UPSI 1
| // EXEC DITTO
| $$DITTO REW INPUT=A61
| $$DITTO TRS INPUT=A61,SCANARG='HDR1'
| $$DITTO EOJ
| /&

```

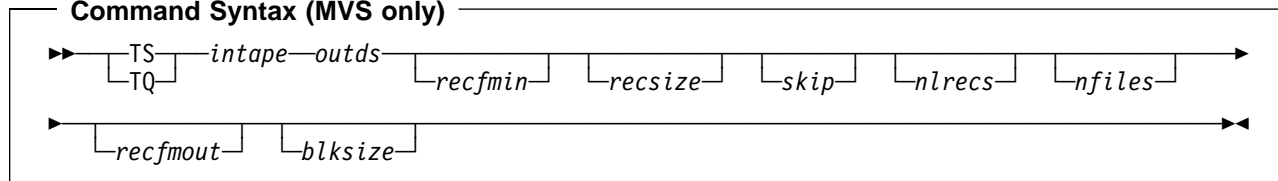
TS, TQ (Tape to Sequential Data)—MVS and VSE Only

Purpose Copy tape records to a sequential data set.

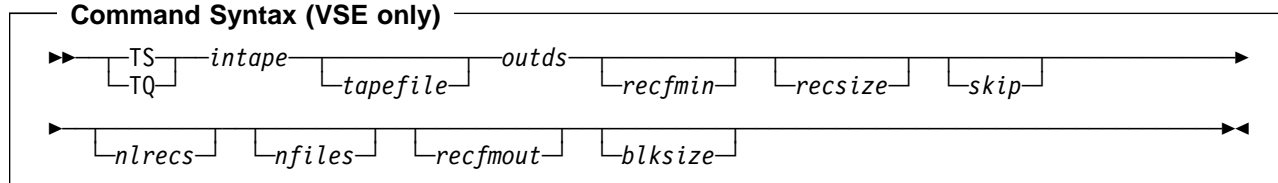
This function can be used to prepare tape data for distribution across networks. For more information, see “Copying Tape Data across a Network” on page 81.

For compatibility with previous releases, you can specify TS or TQ.

Command Syntax (MVS only)



Command Syntax (VSE only)



Batch Syntax

\$\$\$DITTO TS|TQ ...

Keyword Syntax

DITTO \$TS|\$TQ ...

| MVS keywords | VSE keywords | CMS keywords |
|--|--|--------------|
| [LABEL=BLP] INPUT=ddname [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [NFILES={1 EOV EOD nfiles}] [{ OUTPUT={QSAMOUT ddname} DSNOUT=dsname [(member)] [VOLSEROUT=volser] [UNITOUT=unit] [DISP={NEW OLD MOD CAT}] [SPACE=space]]] [RECFMOUT={U UA UM F FA FM FB FBA FBM FBS FBSA FBSM V VA VM VB VBA VBM VS VSA VSM VBS VBSA VBSM D DA DB DBA DS DSA DBS DBSA}] [BLKSIZE=blksize] | INPUT={unit SYSnnn} [{ FILEIN={TAPEIN tlbl} INFILE=tapefile }] [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [NFILES={1 EOV EOD nfiles}] [OUTPUT={unit SYSnnn volser}] [{ FILEOUT={SDSKOUT dlbl} DSNOUT=dsname [VSAMOUT={NO YES}] [BEGIN=begin] [END=end] [UCATOUT=ucate] [DISP={NEW OLD}] [SPACE=space] [AVGSIZE=avgsz]]] [RECFMOUT={U F V FB VB}] [BLKSIZE={MAX blksize}] | |

Parameters

avgsz The average record size for a new VSAM-managed SAM data set.

begin The begin position for a BAM-managed SAM data set in one of the following forms:

cyld Where the last two digits represent the head, and the other digits represent the cylinder (CKD)

Rnnnnnn To indicate the relative track number (CKD)

nnnnnnn To indicate the physical block number (FBA)

| | |
|----------------|--|
| <i>blksize</i> | <p>MVS: maximum length of an output block in bytes. Specify it to change the block size of an old data set, or to override any system default for a new data set.</p> <p>VSE: the actual block size for record format FB, otherwise the maximum block size. For record format FB or VB a block size is required, otherwise it is optional. Specify MAX to use the maximum block size allowed for the output device.</p> |
| <i>ddname</i> | Refers to a DD or TSO ALLOC statement. The default is QSAMOUT for output. |
| <i>disp</i> | Disposition of the output data set. |
| | <p>MVS: specify OLD, MOD, NEW, or CAT for NEW,CATLG. The default is NEW.</p> <p>VSE: for VSAM-managed SAM, specify OLD or NEW. The default is NEW.</p> |
| <i>dlbl</i> | Refers to a DLBL statement. The default is SDSKOUT. |
| <i>dsname</i> | Name of a sequential data set. If specified, any MVS DD or VSE DLBL statements provided are not used. To further describe the data set, use the appropriate keywords as shown in the table. |
| <i>end</i> | <p>The end position for a BAM-managed SAM data set in one of the following forms:</p> <p><i>cylhd</i> Where the last two digits represent the head, and the other digits represent the cylinder (CKD)</p> <p><i>Rnnnnnn</i> To indicate the relative track number (CKD)</p> <p><i>nnnnnnn</i> To indicate the physical block number (FBA)</p> <p><i>Nnnnnnnn</i> To indicate the number of tracks (CKD) or blocks (FBA) to be allocated</p> |
| <i>inout</i> | Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing). |
| <i>intape</i> | Specification of the input tape unit. (Command Syntax only.) |
| | <p>MVS:</p> <ul style="list-style-type: none"> • If the tape is already allocated specify: <i>ddname</i>. • If label processing is desired specify: <i>ddname,tapevol,unit,{SL AL},tapedsn,[seqnum],[xdate],[inout]</i>. • For NL or BLP tape processing specify: <i>ddname,tapevol,unit,{NL BLP},[seqnum],[xdate],[inout]</i>. <p>VSE:</p> <ul style="list-style-type: none"> • Specify: <i>unit[-mm]</i>. |
| LABEL=BLP | <p>Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the <i>DITTO/ESA Installation and Customization Guide</i>.</p> |
| <i>member</i> | For a partitioned data set, the member name. |
| <i>mm</i> | A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67. |
| <i>nfiles</i> | Number of tape files to be copied. The maximum is 99999; the default is 1. |
| | <p>EOV Copies files until a double tape mark or an EOV label is reached.</p> <p>EOD Copies files until the delimiter specified in the EOD parameter of the SET function is reached.</p> <p>Use this parameter to copy a multifile tape, including tape marks, to a single sequential data set. (You can use the ST function to copy the data set back to tape, including the tape marks.) For more information, see “Copying Tape Data across a Network” on page 81.</p> <p>VSE and CMS: for labeled tape processing, only one file is processed. Specifying <i>nfiles</i> is ignored.</p> |

VSE: for labeled tape processing, only one file is processed. Specifying *nfiles* is ignored.

nlrecs Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.

outds Specification of the output data set. (Command Syntax only.)

MVS:

- If a JCL ddname or TSO ALLOC statement for QSAMOUT is provided, no input is required.
- If the data set is cataloged, specify: *dsname*[(*member*)],,{**OLD|MOD**}
- If the data set is not cataloged, specify: *dsname*[(*member*)],[,*unit*],{**OLD|MOD**}
- For a new data set, specify: *dsname*,*volser*[,*unit*],{**NEW|CAT|MOD**},*space*

SMS may modify the allocation of new data sets on your system. See your SMS Administrator for additional information.

VSE:

- If a DLBL statement for SDSKOUT is provided, no input is required.
- For an existing VSAM-managed SAM, specify: *dsname*,**YES**,[*ucaf*],**OLD**
- For a new VSAM-managed SAM, specify: *dsname*,**YES**,[*ucaf*],**NEW**,*space*,*avgsz*
- For a BAM-managed SAM, specify: *dsname*,**NO**,[*unit*|*volser*],*begin*,*end*

recfmin Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

B Blocked
D Variable-length ISO/ANSI tape records
F Fixed length
S Spanned format
U Undefined length
V Variable length

recfmout Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

U Undefined length
F Fixed length
V Variable length
D Variable-length ISO/ANSI tape records
B Blocked
S Standard format (fixed length) or spanned format (variable length)
A Containing ISO/ANSI control characters
M Containing machine code control characters

MVS: use *recfmout* to override existing DCB information. You cannot create a PDS with a record format of FBS. If output data sets with a record format containing the letter D are written to disk, unpredictable results may occur.

recsize Length of the input records, if *recfmin* is F or FB. If *recfmin* is F, the length of the first record is used by default. If *recfmin* is FB, *recsize* is required.

seqnum Relative position of the data set on the tape.

skip Number of logical records to be skipped. The default is 0.

space The number of space units to be allocated for a new data set. Specify *n* or (*n,m*), where *n* is the number of primary units, and *m* is the number of secondary units.

MVS: The space unit is a track.

VSE: The space unit is the length of an average record.

| | |
|-----------------|---|
| <i>SYSnnn</i> | Programmer logical unit to which the input or output device is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapefile</i> | File ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted. |
| <i>tlbl</i> | Refers to a TLBL statement to process a labeled tape. The default is TAPEIN. |
| <i>ucat</i> | For VSAM-managed SAM, the data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE: Device number (3 or 4 hexadecimal digits). |
| <i>volser</i> | Volume serial number for a new or non-cataloged data set. |
| <i>vsamout</i> | Specify YES for VSAM-managed SAM. The default is NO. |
| <i>Vvolser</i> | The letter V followed by the serial number of the output volume. |
| <i>xdate</i> | Expiration date in the form [yy]yy.ddd. |

Batch Example (MVS)

```

//TS JOB (acct),'name' Create Tape Image File
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DITTO EXEC DTOBAT
//SYSPRINT DD SYSOUT=*
//TAPE DD UNIT=(561,,DEFER),VOL=SER=DIT001,
// DISP=SHR,LABEL=(,BLP)
//XMIT DD DSN=KEISTEW.TAPE.XMIT.WORK,
// DISP=(,CATLG),
// UNIT=SYSALLDA,
// SPACE=(CYL,(1,1),RLSE),
// VOL=SER=MVS1T3
//SYSIN DD *
$$DITTO TS INPUT=TAPE,LABEL=BLP,
$$DITTO OUTPUT=XMIT,NFILES=EOV
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB TS - Create Tape Image File
// DLBL OLDXMIT,'TAPE.XMIT.WORK',,VSAM, X
// DISP=(OLD,DELETE)
// DLBL XMIT,'TAPE.XMIT.WORK',,VSAM,DISP=NEW, X
// RECORDS=(100,100),RECSIZE=100
// UPSI 1
// EXEC DITTO
$$DITTO SP FILEIN=OLDXMIT delete old file
$$DITTO REW INPUT=A61 for security only
$$DITTO TS INPUT=A61,FILEOUT=XMIT,
$$DITTO RECFMOUT=U,NFILES=EOV
$$DITTO EOJ
/&

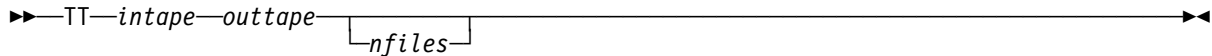
```

TT (Tape to Tape)

Purpose Copy tape files from one tape to another tape (or more than one tape, if necessary). You can specify the number of files to be copied, or you can copy the entire tape. For each tape file copied, block count statistics are printed or displayed. Records are always copied as physical blocks, regardless of whether they are labels or ordinary data. Any leading tape mark is copied.

For further information on tape copy functions, see “Copying from Tape to Tape” on page 80.

Command Syntax



Batch Syntax

`$$DITTO TT ...`

Keyword Syntax

`DITTO $TT ...`

| MVS keywords | VSE keywords | CMS keywords |
|---|---|---|
| <code>[LABEL=BLP]</code> <code>INPUT=ddname</code> <code>OUTPUT=ddname</code> <code>[DENS=mm]</code> <code>[NFILES={1 EOV EOT EOD nfiles}]</code> | <code>INPUT={unit SYSnnn}</code> <code>OUTPUT={unit SYSnnn}</code> <code>[DENS=mm]</code> <code>[NFILES={1 EOV EOT EOD nfiles}]</code> | <code>INPUT=unit</code> <code>OUTPUT=unit</code> <code>[DENS=mm]</code> <code>[NFILES={1 EOV EOT EOD nfiles}]</code> |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
`ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout]`.
- For NL or BLP tape processing specify:
`ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout]`.

VSE or CMS:

- Specify: `unit[-mm]`.

`LABEL=BLP` Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

nfiles Number of tape files to be copied. The maximum is 99 999; the default is 1.

EOV Copies files until a double tape mark or an EOV label is reached.

EOT Copies files until the physical end of the volume

EOD Copies files until the delimiter specified in the EOD parameter of the SET function is reached.

The records from the current tape position up to and including the next tape mark make up one file. However, if the tape is positioned immediately before a tape mark, the file is considered to end at the next tape mark.

outtape Specification of the output tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname[-mm]*.
- If label processing is desired specify:
ddname[-mm],[tapevol],unit,{SL|AL},tapedsn,[seqnum],[xdate].
- For NL or BLP tape processing specify:
ddname[-mm],[tapevol],unit,{NL|BLP},[seqnum],[xdate].

VSE or CMS:

- Specify: *unit[-mm]*.

seqnum Relative position of the data set on the tape.

SYSnnn Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or VOL1) of the tape volume to be mounted, or blank for a non-specific volume request. For a volume set, specify up to five volsers in the form (volser1, volser2...).

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.

VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form *[yy]yy.ddd*.

Batch Example (MVS)

```
//TT JOB (acct),'name' Tape to Tape Copy
//DITTO EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//TAPE1 DD UNIT=381,VOL=SER=DITTO1,
// DISP=OLD,LABEL=(,BLP)
//TAPE2 DD UNIT=382,VOL=SER=DITTO2,
// DISP=OLD,LABEL=(,BLP)
//SYSIN DD *
$$DITTO TT INPUT=TAPE1,OUTPUT=TAPE2,
$$DITTO NFILES=EOV
$$DITTO EOJ
//
```

Batch Example (VSE)

```
// JOB TT - ASCII Tape to EBCDIC Tape Copy
// ASSGN SYS021,991,C0
// UPSI 1
// EXEC DITTO
$$DITTO REW INPUT=SYS021
$$DITTO REW OUTPUT=A62
$$DITTO SET ASCII=IN
$$DITTO TT INPUT=SYS021,OUTPUT=A62
$$DITTO EOJ
/&
```

TTC (Tape to Tape Compare)

Purpose Compare, byte by byte, the data on one tape with the data on another. Comparison starts at the current position of each tape. When the first difference is found, comparison stops and the comparison results are printed.

Command Syntax

```
▶—TTC—intape—outtape—nfiles—▶
```

Batch Syntax

```
$$DITTO TTC ...
```

Keyword Syntax

```
DITTO $TTC ...
```

| MVS keywords | VSE keywords | CMS keywords |
|---|--|--|
| [LABEL=BLP] INPUT= <i>ddname</i> OUTPUT= <i>ddname</i> [NFILES={ <u>1</u> EOV EOD <i>nfiles</i> }] | INPUT={ <i>unit</i> SYS <i>nnn</i> } OUTPUT={ <i>unit</i> SYS <i>nnn</i> } [NFILES={ <u>1</u> EOV EOD <i>nfiles</i> }] | INPUT= <i>unit</i> OUTPUT= <i>unit</i> [NFILES={ <u>1</u> EOV EOD <i>nfiles</i> }] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the first input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inoutf].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inoutf].

VSE or CMS:

- Specify: *unit[-mm]*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
 | the first DITTO function that uses the tape. For BLP processing requirements, see
 | “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization*
 | *Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

nfiles Number of tape files to be compared. The maximum is 99999; the default is 1.

EOV Compares files until a double tape mark or an EOV label is reached.

EOD Compares files until the delimiter specified in the EOD parameter of the SET function is reached.

outtape Specification of the second input tape unit (as described under *intape*).

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or VOL1) of the tape volume to be mounted.

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
 VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form [yy]yy.ddd.

Batch Example (MVS)

```
//TTC JOB (acct),'name' Tape to Tape Compare
//DITTO EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//TAPE1 DD UNIT=381,VOL=SER=DITTO1,
// DISP=OLD,LABEL=(,BLP)
//TAPE2 DD UNIT=382,VOL=SER=DITTO2,
// DISP=OLD,LABEL=(,BLP)
//SYSIN DD *
$$DITTO TTC INPUT=TAPE1,OUTPUT=TAPE2,
$$DITTO NFILES=EOV
$$DITTO EOJ
//
```

Batch Example (VSE)

```
// JOB TTC - Tape to Tape Compare
// UPSI 1
// EXEC DITTO
$$DITTO REW INPUT=A61
$$DITTO REW OUTPUT=A62
$$DITTO TTC INPUT=A61,
$$DITTO OUTPUT=A62,NFILES=3
$$DITTO EOJ
/&
```

TTR (Tape to Tape Reblocked)

Purpose Copy one tape file, starting from the current tape position, to one or more output tapes. You can change the record format and reblock the file during copying. Any leading tape mark is copied.

For further information about tape copy functions, see “Copying from Tape to Tape” on page 80.

Command Syntax

```

TTR intape outtape [recfmin] [recsize] [recfmout] [blksize]
  
```

Batch Syntax

```

$$DITTO TTR ...
  
```

Keyword Syntax

```

DITTO $TTR ...
  
```

| MVS keywords | VSE keywords | CMS keywords |
|--|---|---|
| [LABEL=BLP] INPUT=ddname [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] OUTPUT=ddname [DENS=mm] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] | INPUT={unit SYSnnn} [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] OUTPUT={unit SYSnnn} [DENS=mm] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] | INPUT=unit [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] OUTPUT=unit [DENS=mm] [RECFMOUT={U F FB V VB VBS VS D DB DBS DS}] [BLKSIZE=blksize] |

Parameters

blksize If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

| | |
|-----------------|--|
| <i>outtape</i> | Specification of the output tape unit. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If the tape is already allocated specify: <i>ddname[-mm]</i>. • If label processing is desired specify: <i>ddname[-mm],[tapevol],unit,{SL AL},tapedsn,[seqnum],[xdate]</i>. • For NL or BLP tape processing specify: <i>ddname[-mm],[tapevol],unit,{NL BLP},[seqnum],[xdate]</i>. VSE or CMS: <ul style="list-style-type: none"> • Specify: <i>unit[-mm]</i>. |
| <i>recfmin</i> | Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <p>B Blocked D Variable-length ISO/ANSI tape records F Fixed length S Spanned format U Undefined length V Variable length</p> |
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <p>U Undefined length F Fixed length V Variable length D Variable-length ISO/ANSI tape records B Blocked S Spanned format</p> |
| <i>recsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>recsize</i> is required. |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted, or blank for a non-specific volume request. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form <i>[yy]yy.ddd</i> . |

TTR

Batch Example (MVS)

```
//TTR JOB (acct),'name' Tape to Tape Reblocked
//DITTO EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//TAPE1 DD UNIT=381,VOL=SER=DITTO1,
// DISP=OLD,LABEL=(,BLP)
//TAPE2 DD UNIT=382,VOL=SER=DITTO2,
// DISP=OLD,LABEL=(,BLP)
//SYSIN DD *
$$DITTO TTR INPUT=TAPE1,RECFMIN=VB,
$$DITTO OUTPUT=TAPE2,RECFMOUT=U
$$DITTO EOJ
//
```

Batch Example (VSE)

```
// JOB TTR - Tape to Tape Reblocked
// UPSI 1
// EXEC DITTO
$$DITTO REW INPUT=A61 goto begin of tape
$$DITTO REW OUTPUT=A62 goto begin of tape
$$DITTO TTR INPUT=A61,RECFMIN=FB,RECSIZE=80,
$$DITTO OUTPUT=A62,RECFMOUT=VB,BLKSIZE=1000
$$DITTO EOJ
/&
```


TU (Tape Update)

Purpose Update tape records at the terminal, while copying a tape file to another tape. (If you make a mistake, you still have the original tape file as a backup.) This function is available only in full-screen mode.

For more information about the TU function, use the online help or see “Updating Tape Blocks in Full-Screen Mode” on page 74.

TV (Tape to VSAM)

Purpose Copy tape records to a VSAM data set.

This function can be used to prepare tape data for distribution across networks. For more information, see “Copying Tape Data across a Network” on page 81.

CMS: CMS/VSAM is required.

Command Syntax (MVS only)

```

▶ TV—intape—outds—reuse—recfmin—recsize—skip—nlrecs—
nlfiles—

```

Command Syntax (VSE and CMS only)

```

▶ TV—intape—tapefile—outds—reuse—recfmin—recsize—skip—
nlrecs—nlfiles—

```

Batch Syntax

```
$$$DITTO TV ...
```

Keyword Syntax

```
DITTO $TV ...
```

| MVS keywords | VSE keywords | CMS keywords |
|---|---|---|
| [LABEL=BLP] INPUT=ddname [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [NFILES={1 EOV EOD nfiles}] [{ OUTPUT={VSAMOUT ddname} DSNOUT=dsname}] [REUSE={NO YES}] | INPUT={unit SYSnnn} [{ FILEIN={TAPEIN tlbl} INFILE=tapefile}] [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [NFILES={1 EOV EOD nfiles}] [{ FILEOUT={VDSKOUT dlbl} DSNOUT=dsname UCATOUT=ucat}] [REUSE={NO YES}] [PWDOUT=pwdout] | INPUT=unit [INFILE=tapefile] [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] [NFILES={1 EOV EOD nfiles}] DSNOUT=dsname [UCATOUT=ucat] [REUSE={NO YES}] [PWDOUT=pwdout] |

Parameters

ddname Refers to a DD or TSO ALLOC statement. The default is VSAMOUT for output.

dlbl Refers to a DLBL statement. The default is VDSKOUT.

dsname Name of a VSAM data set.

| | |
|----------------|---|
| <i>inout</i> | Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing). |
| <i>intape</i> | Specification of the input tape unit. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If the tape is already allocated specify: <i>ddname</i>. • If label processing is desired specify: <i>ddname,tapevol,unit,{SL AL},tapedsn,[seqnum],[xdate],[inout]</i>. • For NL or BLP tape processing specify: <i>ddname,tapevol,unit,{NL BLP},[seqnum],[xdate],[inout]</i>. VSE or CMS: <ul style="list-style-type: none"> • Specify: <i>unit[-mm]</i>. |
| LABEL=BLP | Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see "Customizing the Security Environment" in the <i>DITTO/ESA Installation and Customization Guide</i> . |
| | |
| | |
| <i>mm</i> | A 2-byte tape mode code, as shown in "Tape Density and Mode Values" on page 67. |
| <i>nfiles</i> | Number of tape files to be copied. The maximum is 99 999; the default is 1. EOV Copies files until a double tape mark or an EOV label is reached. EOD Copies files until the delimiter specified in the EOD parameter of the SET function is reached. Use this parameter to copy a multifile tape, including tape marks, to a single VSAM entry-sequenced data set. (You can use the VT function to copy the VSAM file back to tape, including the tape marks.) For more information, see "Copying Tape Data across a Network" on page 81. VSE and CMS: for labeled tape processing, only one file is processed. Specifying <i>nfiles</i> is ignored. |
| <i>nrecs</i> | Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>outds</i> | Specification of the VSAM output data set. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If a DD or TSO ALLOC statement for VSAMOUT is provided, no input is required. • If no DD statement is provided specify: <i>dsname</i>. VSE: <ul style="list-style-type: none"> • If a DLBL statement for VDSKOUT is provided, no input is required. • If no DLBL statement is provided specify: <i>dsname,[ucaf]</i>. CMS: <ul style="list-style-type: none"> • Specify: <i>dsname,[ucaf]</i>. |
| <i>pwdout</i> | Password of a password-protected VSAM data set. |
| <i>recfmin</i> | Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <ul style="list-style-type: none"> B Blocked D Variable-length ISO/ANSI tape records F Fixed length S Spanned format U Undefined length |

V Variable length

| | |
|-----------------|--|
| <i>recsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>recsize</i> is required. |
| <i>reuse</i> | Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO. |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>skip</i> | Number of logical records to be skipped. The default is 0. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapefile</i> | VSE: the file ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed. CMS: the file ID of a tape file, if you want to perform label processing. By default, NL processing is performed. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted. |
| <i>tlbl</i> | Refers to a TLBL statement to process a labeled tape. The default is TAPEIN. |
| <i>ucat</i> | The data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form [yy]yy.ddd. |

Keyword Example (MVS)

```

| /* REXX */
| /* copy a tape file to a vsam data set */
| /* allocate input tape */
| /* use SL processing. required file is at */
| /* position 1 */
|
| "ALLOC FILE(TAPE) DA('AASTLE.TAPEOUT'),
|     "VOLUME(DIT001) UNIT(CART) LABEL(SL)",
|     "POSITION(1) OLD"
|
| /* copy a tape file to a vsam data set */
|
| "DITTO $TV INPUT=TAPE,",
|     "DSNOUT=VSAM.DISK.VT55.DATA"
|
| /* free the tape unit */
|
| "FREE FILE(TAPE)"
|
| exit

```

Keyword Example (CMS)

```

/* REXX */
/* Copy a tape file to a VSAM data set */
/* Define the VSAM catalog to work with */
"DITTO $VMC CATALOG=VSESP.USER.CATALOG,",
    "UNIT=14C"
/* Position to the beginning of the tape */
"DITTO $REW INPUT=A60"
/* copy a tape file to a VSAM data set */
"DITTO $TV INPUT=A60,DSNOUT='PRIVATE.WORK' "
/* Unload the tape */
"DITTO $RUN INPUT=A60"
exit

```

TX (Tape to REXX Variable)

Purpose Copy tape records into a REXX stem variable. Each record is copied to a variable named *stem.nnn*. *stem.0* is a counter.

For example, if you copy 3 records, *stem* is VARXX. and VARXX.0 has the value 5:

- The records are copied to VARXX.6, VARXX.7, and VARXX.8.
- The value of VARXX.0 is reset to 8.

This function is available only from a REXX procedure.

Command Syntax (MVS only)

```
TX—intape—stem—recfmin—recsize—skip—nlrecs
```

Command Syntax (VSE and CMS only)

```
TX—intape—tapefile—stem—recfmin—recsize—skip—nlrecs
```

Keyword Syntax

DITTO \$TX ...

| MVS keywords | VSE keywords | CMS keywords |
|--|---|--|
| [LABEL=BLP] INPUT=ddname VARNAME=stem [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] | INPUT={unit SYSnnn} [{ FILEIN={TAPEIN tbl} INFILE=tapefile }] VARNAME=stem [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] | INPUT=unit [INFILE=tapefile] VARNAME=stem [RECFMIN={U F FB V VB VBS VS D DB DBS DS}] [RECSIZE=recsize] [POSITION={0 skip}] [NLRECS={ALL nlrecs}] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

inout Specify YES to use DCB parameter OPEN=INOUT (open for input/output processing).

intape Specification of the input tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*.
- If label processing is desired specify:
ddname,tapevol,unit,{SL|AL},tapedsn,[seqnum],[xdate],[inout].
- For NL or BLP tape processing specify:
ddname,tapevol,unit,{NL|BLP},[seqnum],[xdate],[inout].

VSE or CMS:

- Specify: *unit[-mm]*.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
 | the first DITTO function that uses the tape. For BLP processing requirements, see
 | "Customizing the Security Environment" in the *DITTO/ESA Installation and Customization*
 | *Guide*.

| | |
|-----------------|---|
| <i>mm</i> | A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67. |
| <i>nlrecs</i> | Number of records to be copied or ALL. The maximum number is 99999999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>recfmin</i> | Record format for the input. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: <ul style="list-style-type: none"> B Blocked D Variable-length ISO/ANSI tape records F Fixed length S Spanned format U Undefined length V Variable length |
| <i>recsize</i> | Length of the input records, if <i>recfmin</i> is F or FB. If <i>recfmin</i> is F, the length of the first record is used by default. If <i>recfmin</i> is FB, <i>recsize</i> is required. |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>skip</i> | Number of logical records to be skipped. The default is 0. |
| <i>stem</i> | A REXX stem variable name. The maximum length is 44 characters. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapefile</i> | VSE: the file ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed. CMS: the file ID of a tape file, if you want to perform label processing. By default, NL processing is performed. |
| <i>tapevol</i> | Serial number (external label or V0L1) of the tape volume to be mounted. |
| <i>tlbl</i> | Refers to a TLBL statement to process a labeled tape. The default is TAPEIN. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form <i>[yy]yy.ddd</i> . |

Keyword Example (CMS)

```

/* REXX - copy SL tape file to rexx stem */
"DITTO $REW INPUT 181" /* goto begin of tape */
"DITTO $TX INPUT 181 INFILE TEST.DATA" ,
  "VARNAME TAPE."
Do i=1 To tape.0
  Say tape.i      /* show tape record      */
End
Exit

```

VB (VSAM Browse)

Purpose Browse a VSAM data set at the terminal. This function is available only in full-screen mode.

For more information about the VB function, use the online help or see “Browsing Data” on page 32.

CMS: CMS/VSAM is required.

VC (VSAM to Card)

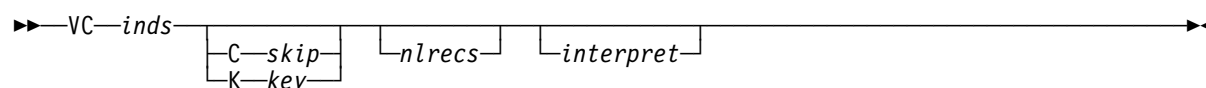
Purpose Copy VSAM records to cards. Output is written to:

- MVS: SYSPUNCH
- VSE: SYSPCH
- CMS: the virtual punch address 000.

The VSAM records must have a record length of 80, 81 (VSE only), or a multiple of 80.

CMS: CMS/VSAM is required.

Command Syntax



Batch Syntax

\$\$DITTO VC ...

Keyword Syntax

DITTO \$VC ...

| MVS keywords | VSE keywords | CMS keywords |
|---|---|--|
| <pre> [{ INPUT={VSAMIN ddname} DSNIN=dsname }] [{ POSITION={0 skip} KEY=key}] [NLRECS={ALL nlracs}] [INTERPRET={NO YES}] </pre> | <pre> [{ FILEIN={VDSKIN dlbl} DSNIN=dsname [UCATIN=ucate] }] [PWDIN=pwdin] [{ POSITION={0 skip} KEY=key}] [NLRECS={ALL nlracs}] [INTERPRET={NO YES}] </pre> | <pre> DSNIN=dsname [UCATIN=ucate] [PWDIN=pwdin] [{ POSITION={0 skip} KEY=key}] [NLRECS={ALL nlracs}] [INTERPRET={NO YES}] </pre> |

Parameters

ddname Refers to a DD or TSO ALLOC statement. The default is VSAMIN.

dlbl Refers to a DLBL statement. The default is VDSKIN.

dsname Name of a VSAM data set.

inds Specification of the VSAM input data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for VSAMIN is provided, no input is required.
- If no DD statement is provided specify: *dsname*.

VSE:

- If a DLBL statement for VDSKIN is provided, no input is required.
- If no DLBL statement is provided specify: *dsname,[ucatl]*.

CMS:

- Specify: *dsname,[ucatl]*.

| | |
|------------------|--|
| <i>interpret</i> | Interpret the output. If you specify YES, the output device must be a card punch with the card print feature installed. The default is NO. |
| <i>key</i> | A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to <i>key</i> is the first record copied. If you omit the <i>key</i> and <i>skip</i> values, copying begins with the first record in the data set. If the key contains lowercase characters, blanks, or commas, enclose it in single quotation marks. You can also specify a key in hexadecimal format, for example: X'C1C2C3'. |
| <i>nlrecs</i> | Number of cards to be punched or ALL. The maximum number is 99999999. If you specify ALL or omit the parameter, all the remaining records are punched. |
| <i>pwdin</i> | Password of a password-protected VSAM data set. |
| <i>skip</i> | Number of card images to be skipped from the beginning of the data set. If you omit the <i>skip</i> and <i>key</i> values, copying begins with the first record in the data set. |
| <i>ucatl</i> | The data set name of the user catalog. |

Batch Example (MVS)

```

//VC JOB (acct),'name' VSAM to Card
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//
//*
//DITTO EXEC DTOBAT
//SYSPUNCH DD SYSOUT=*
$$DITTO VC DSNIN=KEISTEW.VSAM80.FILE
$$DITTO EOJ
//

```

Batch Example (VSE)

```

// JOB VC - VSAM to Card
// UPSI 1
// EXEC DITTO
$$DITTO VC DSNIN=DTT.ITT.RRDS,
$$DITTO UCATIN=DTT.ITT.UCAT1
$$DITTO EOJ
/&

```

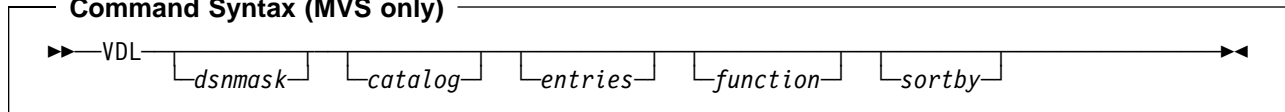
VDL (VSAM Data Set List)

Purpose Work with a list of catalog entries for VSAM data sets. In full-screen mode, you can perform various DITTO functions. In command mode under MVS or CMS, you can invoke the full-screen mode VDL function. In any other case, you can only print a list of catalog entries.

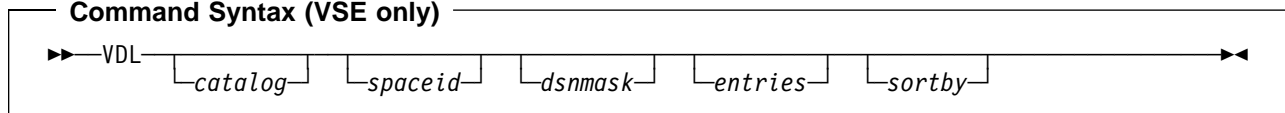
For more information about using this function in full-screen mode, see Chapter 5, "Processing Catalog Entries" on page 51.

CMS: CMS/VSAM is required.

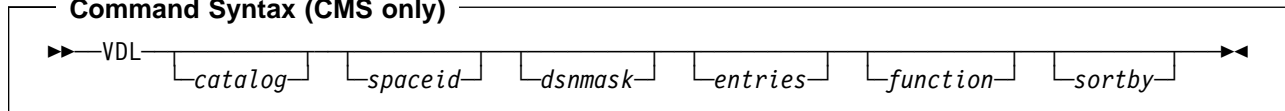
Command Syntax (MVS only)



Command Syntax (VSE only)



Command Syntax (CMS only)



Batch Syntax

\$\$DITTO VDL ...

Keyword Syntax

DITTO \$VDL ...

| MVS keywords | VSE keywords | CMS keywords |
|--|--|--|
| [DSNAME={** <i>dsnmask</i> }] [CATALOG= <i>catalog</i>] [ENTRIES={ALL AIX CLUSTER DATA INDEX PATH UCAT}] [FUNCTION={LIST ALTER DEFINE DELETE PRINT}] [SORTBY={NAME DATE EDATE FREESP}] | [{CATALOG= <i>catalog</i> FILEIN= <i>d1bl</i> }] [DSNAME={** <i>dsnmask</i> }] [VOLSER= <i>spaceid</i>] [ENTRIES={ALL AIX CLUSTER DATA INDEX PATH SPACE UCAT}] [PWDIN= <i>pwdin</i>] [SORTBY={NAME DATE EDATE FREESP EXTENT}] | [CATALOG= <i>catalog</i>] [VOLSER= <i>spaceid</i>] [DSNAME={** <i>dsnmask</i> }] [ENTRIES={ALL AIX CLUSTER DATA INDEX PATH SPACE UCAT}] [FUNCTION={LIST PRINT}] [PWDIN= <i>pwdin</i>] [SORTBY={NAME DATE EDATE FREESP EXTENT}] |

Parameters

catalog Name of a catalog.

MVS: by default, the system catalog search order is used.

VSE or CMS: the default is the master catalog.

d1bl Refers to a DLBL statement.

dsnmask A generic data set name, or search pattern. Within the data set name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) within any number of qualifiers. The default is **.

Under MVS in any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes. For example, the default of ** lists all entries in batch mode; in any other mode, it lists all entries that have the user's TSO prefix as high-level qualifier.

entries The entries you want listed:

ALL All VSAM catalog entries (the default)
AIX Alternate index entries
CLUSTER Cluster entries
DATA VSAM data entries
INDEX VSAM index entries
PATH VSAM path entries
SPACE VSE or CMS: VSAM data space entries
UCAT User catalog entries.

function The command that you want to execute:

LIST Display information contained in the selected catalog entries
ALTER MVS: change the information in the catalog about the specified entry
DEFINE MVS: create a new catalog entry
DELETE MVS: delete the specified catalog entry and its related data
PRINT Print information contained in the selected catalog entries.

pwdin Password of the catalog.

sortby One of the following:

NAME Sorts by data set name (the default)
DATE Sorts by creation date
EDATE Sorts by expiration date
FREESP Sorts by free space
EXTENT VSE or CMS: sorts the contents of a VSAM data space by extent.

spaceid The serial number of the volume that contains a VSAM space, if you want to list only the data sets within that data space.

Batch Example (MVS)

```
//VDL JOB (acct),'name' VSAM Data Set List
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//
// PEND
//*
//DITTO EXEC DTOBAT
//SYSIN DD *
$$DITTO VDL DSNAME=CCCA.V2R1.LEVEL%.*.ABJ,
$$DITTO ENTRIES=ALL,
$$DITTO SORTBY=DATE
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB VDL - VSAM Data Set List
// UPSI 1
// EXEC DITTO
$$DITTO VDL CATALOG=VSAM.MASTER.CATALOG,
$$DITTO DSNAME=**
$$DITTO EOJ
/&
```

VE (VSAM Edit)

Purpose Edit a VSAM data set at the terminal. This function is only available in full-screen mode.

For more information about the VE function, use the online help or see Chapter 4, “Editing Data in Full-Screen Mode” on page 39.

CMS: CMS/VSAM is required.

VER (Display Service Level)

Purpose Display the current level of DITTO you are using, together with the latest PTF applied. This function is the same as the LVL (level) function.

Command Syntax

```
▶▶—VER—▶▶
```

Batch Syntax

```
$$DITTO VER
```

Keyword Syntax

```
DITTO $VER
```

Parameters: None.

VF (VSAM to CMS File)—CMS and CMS/VSE Mode Only

Purpose Copy VSAM records to a CMS file.

You can use this function:

- Under CMS, if you have CMS/VSAM.
- In CMS/VSE mode, to copy data from a VSE system to CMS.

| Command Syntax (CMS only) | | |
|---|--------------|---|
| <pre> ┌───┐ │VF──inds──outfile──┐ │ │ │ ┌──append──┐ │ │ │ │ │ │ │ C──skip──┐ │ │ │ K──key──┘ │ │ │ │ ┌──nlrecs──┐ │ │ └──recfmout──┘ │ └──────────────────┘ </pre> | | |
| Batch Syntax (CMS only) | | Keyword Syntax (CMS only) |
| <pre> \$\$\$DITTO VF ... </pre> | | <pre> DITTO \$VF ... </pre> |
| MVS keywords | VSE keywords | CMS keywords |
| | | <pre> DSNIN=dsname [UCATIN=ucate] [PWDIN=pwdin] [{{POSITION={0 skip} KEY=key}}] [NLRECS={ALL nlrecs}] OUTFILE='outfile' [APPEND={NO YES}] [RECFMOUT={V U F}] </pre> |

Parameters

| | |
|-----------------|--|
| <i>append</i> | Specify YES to add output to the end of an existing CMS file. By default, any existing CMS file with the specified file ID is replaced. |
| <i>dsname</i> | Name of a VSAM data set. |
| <i>inds</i> | Specification of the VSAM input data set. (Command Syntax only.) Specify: <i>dsname</i> , <i>[ucate]</i> . |
| <i>key</i> | A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to <i>key</i> is the first record copied. If you omit the <i>key</i> and <i>skip</i> values, copying begins with the first record in the data set. If the key contains lowercase characters, blanks, or commas, enclose it in single quotation marks. You can also specify a key in hexadecimal format, for example: X'C1C2C3'. |
| <i>nlrecs</i> | Number of records to be copied or ALL. The maximum number is 99999999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>outfile</i> | Specifies the CMS output file identifier: <i>filename filetype [filemode]</i> , enclosed in single quotes. The default filemode is A. |
| <i>pwdin</i> | Password of a password-protected VSAM data set. |
| <i>recfmout</i> | Record format for the output: F Checks that all input records are fixed-length, and creates a fixed format output file. V or U Processes the input records without a length check, and creates a variable-format output file (the default). |
| <i>skip</i> | Number of logical records to be skipped from the beginning of the data set. If you omit the <i>skip</i> and <i>key</i> values, copying begins with the first record in the data set. |

ucat The data set name of the user catalog.

VL (VSAM to Library Member)—VSE Only

Purpose Copy VSAM records to a library member.

Command Syntax

```

▶—VL—inds—┬─C—skip─┬─nlrecs─┬─outlib—member─┬─recfmout─▶
              └─K—key—┘
  
```

Batch Syntax

```
$$DITTO VL ...
```

Keyword Syntax

```
DITTO $VL ...
```

| MVS keywords | VSE keywords | CMS keywords |
|--------------|---|--------------|
| | <pre> [{ FILEIN={VDSKIN dlbl} DSNIN=dsname [UCATIN=ucat] }] [PWDIN=pwdin] [{ POSITION={0 skip} KEY=key }] [NLRECS={ALL nlrecs}] LIBOUT=lib MEMBEROUT=member [RECFMOUT={F S}] </pre> | |

Parameters

- dlbl* Refers to a DLBL statement. The default is VDSKIN.
- dsname* Name of a VSAM data set.
- inds* Specification of the VSAM input data set. (Command Syntax only.)
- If a DLBL statement for VDSKIN is provided, no input is required.
 - If no DLBL statement is provided specify: *dsname*,*[ucat]*.
- key* A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, copying begins with the first record in the data set.
- If the key contains lowercase characters, blanks, or commas, enclose it in single quotation marks. You can also specify a key in hexadecimal format, for example: X'C1C2C3'.
- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- member* Name of the library member in the form *name.type*.
- nlrecs* Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied.
- outlib* Specification of the output library. (Command Syntax only.)
- If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucat]*.
- pwdin* Password of a password-protected VSAM data set.

- recfmout* Record format for the output:
- F** Fixed-length, 80 (the default)
 - S** String format (always used if the member type is PHASE).
- skip* Number of logical records to be skipped from the beginning of the data set. If you omit the *skip* and *key* values, copying begins with the first record in the data set.
- ucat* The data set name of the user catalog.
- volser* Volume serial number of the device where the library is located.

Batch Example (VSE)

```

// JOB VL - VSAM to Library Member
// EXEC IDCAMS,SIZE=AUTO
DELETE DITTO.TC.TEMP.VSAM.VL2Q33 CLUSTER -
  PURGE ERASE -
  CATALOG(VSESP.USER.CATALOG)
DEFINE CLUSTER ( -
  NAME( DITTO.TC.TEMP.VSAM.VL2Q33 ) -
  CISZ(4096) -
  FREESPACE(10 10) -
  NONINDEXED -
  RECORDS(900 100) -
  RECORDSIZE(80 80) -
  REUSE -
  VOL(SYSWK1) ) -
  DATA( -
  NAME(DITTO.TC.TEMP.VSAM.VL2Q33.DATA) ) -
  CATALOG(VSESP.USER.CATALOG)
/*
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VDSKOUT,'DITTO.TC.TEMP.VSAM.VL2Q33',0,
      VSAM,,CAT=MYCAT,DISP=(OLD,KEEP)

// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO BV NLRECS=800,RECSIZE=80,FILLCHAR=AN,
$$DITTO      KEYLOC=73
$$DITTO EOJ
/*
// UPSI 1
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VDSKIN,'DITTO.TC.TEMP.VSAM.VL2Q33',0,
      VSAM,,CAT=MYCAT,DISP=(OLD,KEEP)
// DLBL DITTEST,'DITTEST.WORK.LIB',,VSAM,
      CAT=MYCAT

// EXEC DITTO
$$DITTO VER
$$DITTO VL POSITION=12,LIBOUT=DITTEST.LIB3,
$$DITTO      MEMBEROUT=DITSTF.A
$$DITTO EOJ
/*
/&

```

VMC (VSAM Master Catalog Setup)—CMS Only

Purpose Specify the device address and name of the VSAM master catalog. In batch mode, the master catalog definition applies only to the job. In any other mode, the master catalog definition is saved in the session profile, and is used for all future DITTO sessions by the same user ID.

Command Syntax

▶—VMC—*unit*—*catalog*—◀

Batch Syntax

\$\$DITTO VMC ...

Keyword Syntax

DITTO \$VMC ...

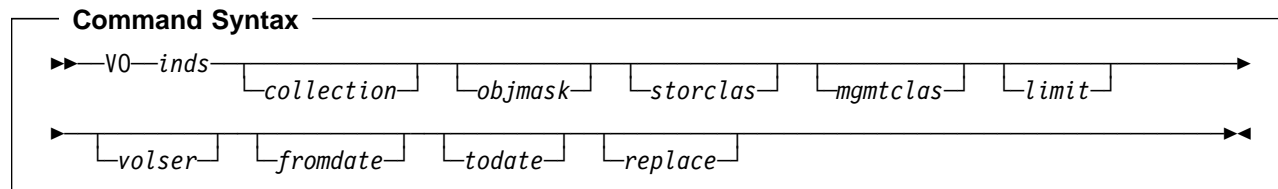
| MVS keywords | VSE keywords | CMS keywords |
|--------------|--------------|--|
| | | CATALOG= <i>catalog</i> UNIT= <i>unit</i> |

Parameters

- catalog* Data set name of the master catalog.
- unit* Device number (3 or 4 hexadecimal digits).

VO (VSAM to Object)—MVS Only

Purpose Restore one or more OAM objects that were backed up in a VSAM data set, or create a new object from a VSAM data set. For more information about using this function, see “Restoring OAM Objects from a Sequential or VSAM Data Set” on page 139.



| Batch Syntax | Keyword Syntax |
|----------------|----------------|
| \$DITTO VO ... | DITTO \$VO ... |

| MVS keywords | VSE keywords | CMS keywords |
|--|--------------|--------------|
| <pre> [{ INPUT={VSAMIN ddname} DSNIN=dsname }] [COLLECTN={** collection}] [OBJNAME={** objmask}] [STORCLAS=storclas] [MGMTCLAS=mgmtclas] [LIMIT={DASD OPTICAL VOLUME}] [VOLSER=volser] [FROMDATE=fromdate] [TODATE=todate] [REPLACE={NO YES}] </pre> | | |

Parameters

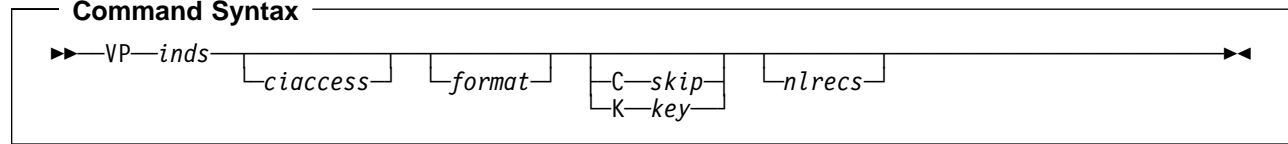
- collection** If the data set contains one object with a header, you can optionally specify a collection name to override the collection name in the header.
- If the data set contains one object with no header (or, equivalently, it contains data that was not previously an object), you must specify the collection where the object will be stored.
- If the data set contains one or more objects that were stored during a multiple-object backup, you can optionally specify a collection name to restore only objects from the specified collection.
- In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.
- ddname** Refers to a DD or TSO ALLOC statement. The default is VSAMIN.
- dsname** Name of a VSAM data set.
- fromdate** The earliest creation date, in the form *yyyymmdd*. By default, all objects are copied regardless of their creation date.
- inds** Specification of the VSAM input data set (Command Syntax only):
- If a DD or TSO ALLOC statement for VSAMIN is provided, no input is required.
 - If no DD statement is provided specify: *dsname*.
- limit** To limit the objects that are copied based on where the objects were located (before they were backed up to a data set), specify one of the following:
- DASD** Only objects that were stored on DASD
- OPTICAL** Only objects that were stored on optical disks

| | |
|-----------------|--|
| | VOLUME Only objects that were stored on a specified optical disk. You must also specify the <i>volser</i> of the optical disk. |
| <i>mgmtclas</i> | Name of an SMS management class. |
| <i>objmask</i> | <p>If the data set contains one object with a header, you can optionally specify an object name to override the object name in the header.</p> <p>If the data set contains one object with no header (or, equivalently, it contains data that was not previously an object), you must specify a name for the object.</p> <p>If the data set contains one or more objects that were stored during a multiple-object backup, you can optionally specify an object name or generic object name to limit the objects that are restored based on object name. Within the name, you can include a percent sign (%) to represent exactly one character, an asterisk (*) to represent any number of characters (or no characters) within a qualifier, or two asterisks (**) to represent any number of characters (or no characters) with any number of qualifiers.</p> <p>In any mode except batch mode, the user's TSO prefix (normally the user ID) is used as the high-level qualifier for any name that is not entered in quotes.</p> |
| <i>replace</i> | Specify YES to replace any existing objects with the same name in the collection. By default, the object is not copied. |
| <i>storclas</i> | Name of an SMS storage class. |
| <i>todate</i> | The latest creation date, in the form <i>yyyymmdd</i> . |
| <i>volser</i> | If you specified VOLUME for <i>limit</i> , the volume serial number of the optical disk. |

VP (VSAM Print)

Purpose Print VSAM records in character or hexadecimal format.

CMS: CMS/VSAM is required.



Batch Syntax
 \$\$DITTO VP ...

Keyword Syntax
 DITTO \$VP ...

| MVS keywords | VSE keywords | CMS keywords |
|---|--|---|
| [{ INPUT={VSAMIN <i>ddname</i> } DSNIN= <i>dsname</i> }] [CIACCESS={NO YES}] [FORMAT={CHAR HEX}] [{POSITION={0 <i>skip</i> } KEY= <i>key</i> }] [NLRECS={ALL <i>nlrecs</i> }] | [{ FILEIN={VDSKIN <i>dLBL</i> } DSNIN= <i>dsname</i> [UCATIN= <i>ucat</i>] }] [PWDIN= <i>pwdin</i>] [CIACCESS={NO YES}] [FORMAT={CHAR HEX}] [{POSITION={0 <i>skip</i> } KEY= <i>key</i> }] [NLRECS={ALL <i>nlrecs</i> }] | DSNIN= <i>dsname</i> [UCATIN= <i>ucat</i>] [PWDIN= <i>pwdin</i>] [CIACCESS={NO YES}] [FORMAT={CHAR HEX}] [{POSITION={0 <i>skip</i> } KEY= <i>key</i> }] [NLRECS={ALL <i>nlrecs</i> }] |

Parameters

ciaccess Specify YES to process control intervals. The default is to process logical records.

ddname Refers to a DD or TSO ALLOC statement. The default is VSAMIN.

dLBL Refers to a DLBL statement. The default is VDSKIN.

dsname Name of a VSAM data set.

format The format of the output:
CHAR Character format (the default)
HEX Hexadecimal format.

inds Specification of the VSAM input data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for VSAMIN is provided, no input is required.
- If no DD statement is provided specify: *dsname*.

VSE:

- If a DLBL statement for VDSKIN is provided, no input is required.
- If no DLBL statement is provided specify: *dsname*, [*ucat*].

CMS:

- Specify: *dsname*, [*ucat*].

key A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record printed. If you omit the *key* and *skip* values, printing begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in single quotation marks. You can also specify a key in hexadecimal format, for example: X'C1C2C3'.

VP

- nlrecs* Number of records or control intervals to be printed. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records (or control intervals) are printed.
- pwdin* Password of a password-protected VSAM data set.
- skip* Number of logical records (or control intervals) to be skipped from the beginning of the data set. If you omit the *skip* and *key* values, printing begins with the first record in the data set.
- ucat* The data set name of the user catalog.

Batch Example (MVS)

```
//VP JOB (acct),'name' VSAM Print
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//DITTO EXEC DTOBAT
//VSAM DD DSN=KEISTEW.CHANGE.ABJ,DISP=SHR
//SYSIN DD *
$$DITTO VP INPUT=VSAM
$$DITTO EOJ
/*
```

Batch Example (VSE)

```
// JOB VP - VSAM Print
// EXEC IDCAMS,SIZE=AUTO
DELETE DITTO.TC.VSAM.VP2D12 CLUSTER PURGE -
ERASE -
CATALOG(VSESP.USER.CATALOG)
DEFINE CLUSTER ( -
NAME( DITTO.TC.VSAM.VP2D12 ) -
CISZ(4096) -
FREESPACE(30 30) -
NONINDEXED -
RECORDS(500 500) -
RECORDSIZE(133 133) -
REUSE -
VOL(SYSWK1) ) -
DATA( NAME(DITTO.TC.VSAM.VP2D12.DATA) ) -
CATALOG(VSESP.USER.CATALOG)
/*
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VDSKOUT,'DITTO.TC.VSAM.VP2D12',0,
VSAM,,CAT=MYCAT,DISP=(OLD,KEEP)
// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO BV NLRECS=10,RECSIZE=133,
$$DITTO FILLCHAR=BIN,KEYLOC=40
$$DITTO EOJ
/*
// UPSI 1
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VDSKIN,'DITTO.TC.VSAM.VP2D12',0,
VSAM,,CAT=MYCAT,DISP=(OLD,KEEP)
// EXEC DITTO
$$DITTO VER
$$DITTO VP FORMAT=HEX
$$DITTO EOJ
/*
/&
```

VRU (VSAM Record Update)

Purpose Update a single record in a VSAM data set.

This function is not available in batch mode or command mode. In full-screen mode, VE is usually more convenient than VRU.

CMS: CMS/VSAM is required.



Parameters

ciaccess Specify YES to process control intervals. The default is to process logical records.

dsname Name of a VSAM data set.

inds Specification of the VSAM input data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for VSAMIN is provided, no input is required.
- If no DD statement is provided specify: *dsname*.

VSE:

- If a DLBL statement for VDSKIN is provided, no input is required.
- If no DLBL statement is provided specify: *dsname*, [*ucat*].

CMS:

- Specify: *dsname*, [*ucat*].

key A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is altered. If you omit the *key* and *skip* values, the first record in the data set is altered.

If the key contains lowercase characters, blanks, or commas, enclose it in single quotation marks. You can also specify a key in hexadecimal format, for example: X'C1C2C3'.

skip Number of logical records (or control intervals) to be skipped from the beginning of the data set. If you omit the *skip* and *key* values, the first record in the data set is altered.

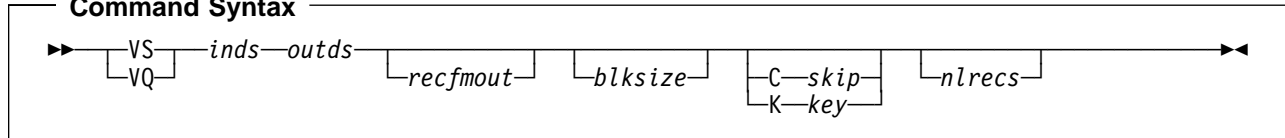
ucat The data set name of the user catalog.

VS, VQ (VSAM to Sequential Data)—MVS and VSE Only

Purpose Copy VSAM records to a sequential data set.

For compatibility with previous releases, you can specify either VS or VQ.

Command Syntax



Batch Syntax

\$\$\$DITTO VS|VQ ...

Keyword Syntax

DITTO \$VS|\$VQ ...

| MVS keywords | VSE keywords | CMS keywords |
|--|---|--------------|
| <pre> [{ INPUT={VSAMIN ddname} DSNIN=dsname }] [{ POSITION={0 skip} KEY=key}] [NLRECS={ALL nlrecs}] [{ OUTPUT={QSAMOUT ddname} DSNOUT=dsname [(member)] [VOLSEROUT=volser] [UNITOUT=unit] [DISP={NEW OLD MOD CAT}] [SPACE=space] }] [RECFMOUT={U UA UM F FA FM FB FBA FBM FBS FBSA FBSM V VA VM VB VBA VBM VS VSA VSM VBS VBSA VBSM D DA DB DBA DS DSA DBS DBSA}] [BLKSIZE=blksize] </pre> | <pre> [{ FILEIN={VDSKIN dlbl} DSNIN=dsname [UCATIN=ucat] }] [PWDIN=pwdin] [{ POSITION={0 skip} KEY=key}] [NLRECS={ALL nlrecs}] [OUTPUT={unit SYSnnn Vvolser}] [{ FILEOUT={SDSKOUT dlbl} DSNOUT=dsname [VSAMOUT={NO YES}] [BEGIN=begin] [END=end] [UCATOUT=ucat] [DISP={NEW OLD}] [SPACE=space] [AVGSIZE=avgsize] }] [RECFMOUT={U F V FB VB}] [BLKSIZE={MAX blksize}] </pre> | |

Parameters

- avgsize* The average record size for a new VSAM-managed SAM data set.
- begin* The begin position for a BAM-managed SAM data set in one of the following forms:
 - cylhd* Where the last two digits represent the head, and the other digits represent the cylinder (CKD)
 - Rnnnnnn* To indicate the relative track number (CKD)
 - nnnnnnn* To indicate the physical block number (FBA)
- blksize* MVS: maximum length of an output block in bytes. Specify it to change the block size of an old data set, or to override any system default for a new data set.
 VSE: the actual block size for record format FB, otherwise the maximum block size. For record format FB or VB a block size is required, otherwise it is optional. Specify MAX to use the maximum block size allowed for the output device.
- ddname* Refers to a DD or TSO ALLOC statement. The default is VSAMIN for input, and QSAMOUT for output.
- disp* Disposition of the output data set.
 MVS: specify OLD, MOD, NEW, or CAT for NEW,CATLG. The default is NEW.
 VSE: for VSAM-managed SAM, specify OLD or NEW. The default is NEW.

| | |
|-----------------|---|
| <i>dlbl</i> | Refers to a DLBL statement. The default is VDSKIN for input, and SDSKOUT for output. |
| <i>dsname</i> | Name of a sequential data set or VSAM data set. If you specify a sequential data set name, any MVS DD or VSE DLBL statements provided are not used. To further describe the sequential data set, use the appropriate keywords as shown in the table. |
| <i>end</i> | The end position for a BAM-managed SAM data set in one of the following forms: <i>cylhd</i> Where the last two digits represent the head, and the other digits represent the cylinder (CKD) <i>Rnnnnnn</i> To indicate the relative track number (CKD) <i>nnnnnnn</i> To indicate the physical block number (FBA) <i>Nnnnnnnn</i> To indicate the number of tracks (CKD) or blocks (FBA) to be allocated |
| <i>inds</i> | Specification of the VSAM input data set. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If a DD or TSO ALLOC statement for VSAMIN is provided, no input is required. • If no DD statement is provided specify: <i>dsname</i>. VSE: <ul style="list-style-type: none"> • If a DLBL statement for VDSKIN is provided, no input is required. • If no DLBL statement is provided specify: <i>dsname,[ucaf]</i>. |
| <i>key</i> | A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to <i>key</i> is the first record copied. If you omit the <i>key</i> and <i>skip</i> values, copying begins with the first record in the data set. If the key contains lowercase characters, blanks, or commas, enclose it in single quotation marks. You can also specify a key in hexadecimal format, for example: X'C1C2C3'. |
| <i>member</i> | For a partitioned data set, the member name. |
| <i>nrecs</i> | Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>outds</i> | Specification of the output data set. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If a JCL ddname or TSO ALLOC statement for QSAMOUT is provided, no input is required. • If the data set is cataloged, specify: <i>dsname[(member)],,{OLD MOD}</i> • If the data set is not cataloged, specify: <i>dsname[(member)],,[unit],{OLD MOD}</i> • For a new data set, specify: <i>dsname,volser[,unit],{NEW CAT MOD},space</i> SMS may modify the allocation of new data sets on your system. See your SMS Administrator for additional information. VSE: <ul style="list-style-type: none"> • If a DLBL statement for SDSKOUT is provided, no input is required. • For an existing VSAM-managed SAM, specify: <i>dsname,YES,[ucaf],OLD</i> • For a new VSAM-managed SAM, specify: <i>dsname,YES,[ucaf],NEW,space,avgsz</i> • For a BAM-managed SAM, specify: <i>dsname,NO,[unit]volser[,begin,end]</i> |
| <i>pwdin</i> | Password of a password-protected VSAM data set. |
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: |

VS

- U** Undefined length
- F** Fixed length
- V** Variable length
- D** Variable-length ISO/ANSI tape records
- B** Blocked
- S** Standard format (fixed length) or spanned format (variable length)
- A** Containing ISO/ANSI control characters
- M** Containing machine code control characters

MVS: use *recfmout* to override existing DCB information. You cannot create a PDS with a record format of FBS. If output data sets with a record format containing the letter D are written to disk, unpredictable results may occur.

- skip* Number of logical records to be skipped from the beginning of the data set. If you omit the *skip* and *key* values, copying begins with the first record in the data set.
- space* The number of space units to be allocated for a new data set. Specify *n* or (*n,m*), where *n* is the number of primary units, and *m* is the number of secondary units.
MVS: The space unit is a track.
VSE: The space unit is the length of an average record.
- SYSnnn* Programmer logical unit to which the output device is assigned.
- ucat* For VSAM or VSAM-managed SAM, the data set name of the user catalog.
- unit* MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
VSE: Device number (3 or 4 hexadecimal digits).
- volser* Volume serial number for a new or non-cataloged data set.
- vsamout* Specify YES for VSAM-managed SAM. The default is NO.
- Vvolser* The letter V followed by the serial number of the output volume.

Batch Example (MVS)

```

//VS JOB (acct),'name' VSAM to QSAM
//*
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
/* EXAMPLE VS JOB
/* ALL RECORDS ARE COPIED FROM THE INPUT
/* VSAM FILE TO THE OUTPUT SEQUENTIAL FILE.
/* THE OUTPUT FILE IS DYNAMICALLY ALLOCATED
/* WITHIN DITTO
//DITTO EXEC DTOBAT
//SYSIN DD *
$$DITTO VS DSNIN=KEISTEW.MESSAGE.ABJ,
* NOTE: DSNIN HAS RECSIZE = 1024
$$DITTO DSNOUT=KEISTEW.MESSAGE.SEQ,
$$DITTO VOLSEROUT=MVS1T3,
$$DITTO DISP=CAT,SPACE=(3,1),
* NOTE: BLKSIZE MUST BE
* 8 BYTES > RECSIZE FOR VB
$$DITTO RECFMOUT=VB,BLKSIZE=1032
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB VS - VSAM to SAM Data
// EXEC IDCAMS,SIZE=AUTO
DELETE DITTO.TC.VSAM.VS262S CLUSTER PURGE -
ERASE -
CATALOG(VSESP.USER.CATALOG)
DEFINE CLUSTER ( -
NAME( DITTO.TC.VSAM.VS262S ) -
CISZ(8192) -
FREESPACE(30 30) -
NONINDEXED -
RECORDS(500 50) -
RECORDSIZE(80 80) -
REUSE -
VOL(SYSWK1) ) -
DATA( NAME(DITTO.TC.VSAM.VS262S.DATA) ) -
CATALOG(VSESP.USER.CATALOG)
/*
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VDSKOUT,'DITTO.TC.VSAM.VS262S',0,
VSAM,,CAT=MYCAT

// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO BV NLRECS=1750,RECSIZE=80,
FILLCHAR=BIN,KEYLOC=1,KEYLEN=4

$$DITTO EOJ
/*
// UPSI 1
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VDSKIN,'DITTO.TC.VSAM.VS262S',0,
VSAM,,CAT=MYCAT
// DLBL SDSKOUT,'DITTO.TC.TEMP.VS1VB',0,
VSAM,,CAT=MYCAT,
DISP=(NEW,KEEP,DELETE),
RECORDS=(900,300),RECSIZE=80

// EXEC DITTO
$$DITTO VER
$$DITTO VS POSITION=100,NLRECS=ALL,
$$DITTO RECFMOUT=VB,BLKSIZE=16000
$$DITTO EOJ
/*
/&
* $$ EOJ

```

VSE (Start DITTO/ESA for VSE)—CMS Only

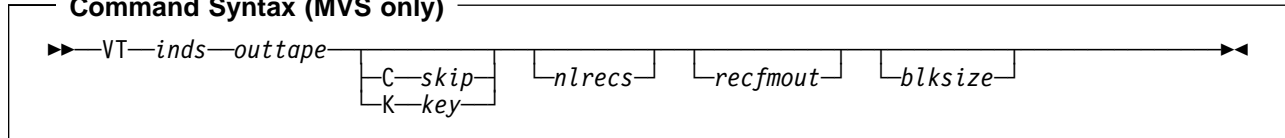
Purpose Invoke DITTO/ESA for VSE in client/server mode. This function is available only in full-screen mode under CMS. For more information about the VSE function, use the online help or see “CMS/VSE Mode: Starting VSE/DITTO from CMS” on page 14.

VT (VSAM to Tape)

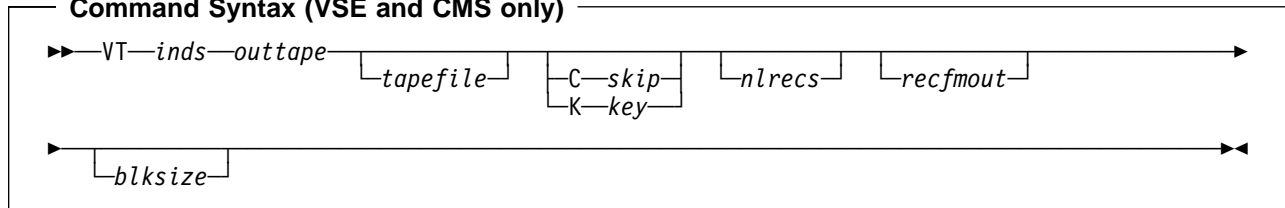
Purpose Copy VSAM records to a tape file.

CMS: CMS/VSAM is required.

Command Syntax (MVS only)



Command Syntax (VSE and CMS only)



Batch Syntax

\$\$\$DITTO VT ...

Keyword Syntax

DITTO \$VT ...

| MVS keywords | VSE keywords | CMS keywords |
|---|--|--|
| <pre>[{ INPUT={VSAMIN ddname} DSNIN=dsname }] [{{ POSITION={0 skip} KEY=key}}] [NLRECS={ALL nlrecs}] [LABEL=BLP] OUTPUT=ddname [DENS=mm] [RECFMOUT={U F FB V VB VBS VS D} DB DBS DS}] [BLKSIZE=blksize]</pre> | <pre>[{ FILEIN={VDSKIN dlbl} DSNIN=dsname UCATIN=ucat }] [PWDIN=pwdin] [{{ POSITION={0 skip} KEY=key}}] [NLRECS={ALL nlrecs}] OUTPUT={unit SYSnnn} [DENS=mm] [{{ FILEOUT={TAPEOUT tbl} OUTFILE=tapefile }}] [RECFMOUT={U F FB V VB VBS VS D} DB DBS DS}] [BLKSIZE=blksize]</pre> | <pre>DSNIN=dsname [UCATIN=ucat] [PWDIN=pwdin] [{{ POSITION={0 skip} KEY=key}}] [NLRECS={ALL nlrecs}] OUTPUT=unit [DENS=mm] [OUTFILE=tapefile] [RECFMOUT={U F FB V VB VBS VS D} DB DBS DS}] [BLKSIZE=blksize]</pre> |

Parameters

blksize If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.

ddname Refers to a DD or TSO ALLOC statement. The default is VSAMIN for input. To create the output tape file with standard labels, specify AL or SL processing on the DD or TSO ALLOC statement.

dlbl Refers to a DLBL statement. The default is VDSKIN.

dsname Name of a VSAM data set.

inds Specification of the VSAM input data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for VSAMIN is provided, no input is required.
- If no DD statement is provided specify: *dsname*.

VSE:

- If a DLBL statement for VDSKIN is provided, no input is required.
- If no DLBL statement is provided specify: *dsname*,*[ucaf]*.

CMS:

- Specify: *dsname*,*[ucaf]*.

key A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, copying begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in single quotation marks. You can also specify a key in hexadecimal format, for example: X'C1C2C3'.

| LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with
| the first DITTO function that uses the tape. For BLP processing requirements, see
| "Customizing the Security Environment" in the *DITTO/ESA Installation and Customization*
| *Guide*.

mm A 2-byte tape mode code, as shown in "Tape Density and Mode Values" on page 67.

nlrecs Number of records to be copied or ALL. The maximum number is 99999999. If you specify ALL or omit the parameter, all the remaining records are copied.

outtape Specification of the output tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname**[-mm]*.
- If label processing is desired specify:
*ddname**[-mm]*,*[tapevol]*,*unit*,**{SL|AL}**,*tapedsn*,*[seqnum]*,*[xdate]*.
- For NL or BLP tape processing specify:
*ddname**[-mm]*,*[tapevol]*,*unit*,**{NL|BLP}**,*[seqnum]*,*[xdate]*.

VSE or CMS:

- Specify: *unit**[-mm]*.

pwdin Password of a password-protected VSAM data set.

recfmout Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters:

U Undefined length
F Fixed length
V Variable length
D Variable-length ISO/ANSI tape records
B Blocked
S Spanned format

seqnum Relative position of the data set on the tape.

skip Number of logical records to be skipped from the beginning of the data set. If you omit the *skip* and *key* values, copying begins with the first record in the data set.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

VT

| | |
|-----------------|--|
| <i>tapefile</i> | VSE: the file ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed. CMS: the file ID of a tape file, if you want to perform label processing. By default, NL processing is performed. |
| <i>tapevol</i> | Serial number (external label or V0L1) of the tape volume to be mounted, or blank for a non-specific volume request. For a volume set, specify up to five volsers in the form (volser1, volser2...). |
| <i>tlbl</i> | Refers to a TLBL statement to process a labeled tape. The default is TAPEOUT. |
| <i>ucat</i> | The data set name of the user catalog. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form [yy]yy.ddd. |

Batch Example (MVS)

```

//VT JOB (acct),'name' VSAM to TAPE
//DTOBAT PROC
//DTOBAT EXEC PGM=DITTO
//STEPLIB DD DSN=DIT.H0GB210.SDITMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//IDCPROC PROC
//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// PEND
//*
//CREATE EXEC IDCPROC,REGION=6M
//SYSIN DD *
DELETE KEISTEW.VSAM.VT55 CLUSTER PURGE ERASE
DEFINE CLUSTER ( -
    NAME( KEISTEW.VSAM.VT55 ) -
    CISZ(4096) -
    FREESPACE(30 30) -
    NONINDEXED -
    KEYS(12 0) -
    RECORDS(1000 100) -
    RECORDSIZE(133 133) -
    REUSE -
    VOL(MVS1W2) ) -
    DATA( NAME(KEISTEW.VSAM.VT55.DATA) )
/*
/**
//STP1005 EXEC DTOBAT
//VSAMIN DD DISP=SHR,DSN=KEISTEW.VSAM.VT55
//TAPE DD UNIT=CART,
// VOL=(,RETAIN,,,SER=DIT001),
// LABEL=(1,SL),DISP=(,KEEP),
// DSN=AASTLE.TAPEOUT
//SYSIN DD *
$$DITTO VER
* GENERATE SOME DATA AND PLACE
* IN A VSAM FILE
$$DITTO BV OUTPUT=VSAMIN,REUSE=NO,NLRECS=4000,
$$DITTO RECSIZE=133,FILLCHAR=BIN,KEYLOC=1
* NOW COPY THE VSAM FILE TO TAPE
$$DITTO VT INPUT=VSAMIN,OUTPUT=TAPE
$$DITTO EOJ
/*

```

Batch Example (VSE)

```

// JOB VT - VSAM to Tape
/. C
// ASSGN SYS010,580
// EXEC IDCAMS,SIZE=AUTO
DELETE DITTO.TC.VSAM.BV2J32 CLUSTER PURGE -
ERASE -
    CATALOG(VSESP.USER.CATALOG)
DEFINE CLUSTER ( -
    NAME( DITTO.TC.VSAM.BV2J32 ) -
    CISZ(4096) -
    FREESPACE(30 30) -
    NONINDEXED -
    RECORDS(1000 100) -
    RECORDSIZE(800 800) -
    REUSE -
    VOL(SYSWK1) ) -
    DATA( NAME(DITTO.TC.VSAM.BV2J32.DATA) ) -
    CATALOG(VSESP.USER.CATALOG)
/*
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VDSKOUT,'DITTO.TC.VSAM.BV2J32',0,
    VSAM,,CAT=MYCAT

// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO BV NLRECS=2455,RECSIZE=800,
$$DITTO FILLCHAR=AN,KEYLOC=1
$$DITTO EOJ
/*
// UPSI 1
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VDSKIN,'DITTO.TC.VSAM.BV2J32',0,
    VSAM,,CAT=MYCAT

// TLBL TAPEOUT,'DIT.VSAM.TAPEOUT',0,DIT001
// EXEC DITTO
$$DITTO VER
$$DITTO VT POSITION=800,NLRECS=1200,
$$DITTO OUTPUT=SYS010,
$$DITTO RECFMOUT=FB,BLKSIZE=8000
$$DITTO EOJ
/*
/&

```

VU (VSAM Update)

Purpose Browse and update a VSAM data set at the terminal. This function is available only in full-screen mode.

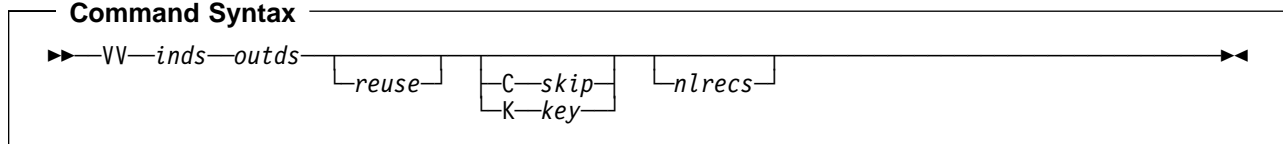
For more information about the VU function, use the online help or see “Updating VSAM Data Sets in Full-Screen Mode” on page 112.

CMS: CMS/VSAM is required.

VV (VSAM to VSAM)

Purpose Copy VSAM records to another VSAM data set.

CMS: CMS/VSAM is required.



Batch Syntax

```
$$DITTO VV ...
```

Keyword Syntax

```
DITTO $VV ...
```

| MVS keywords | VSE keywords | CMS keywords |
|---|--|---|
| <pre>[{ INPUT={VSAMIN ddname} DSNIN=dsname }] [{{POSITION={0 skip} KEY=key}}] [NLRECS={ALL nlrecs}] [{{ OUTPUT={VSAMOUT ddname} DSNOUT=dsname }] [REUSE={NO YES}]</pre> | <pre>[{ FILEIN={VDSKIN dlbl} DSNIN=dsname [UCATIN=ucatl]}] [PWDIN=pwdin] [{{POSITION={0 skip} KEY=key}}] [NLRECS={ALL nlrecs}] [{{ FILEOUT={VDSKOUT dlbl} DSNOUT=dsname [UCATOUT=ucatl]}] [REUSE={NO YES}] [PWDOUT=pwdout]</pre> | <pre>DSNIN=dsname [UCATIN=ucatl] [PWDIN=pwdin] [{{POSITION={0 skip} KEY=key}}] [NLRECS={ALL nlrecs}] DSNOUT=dsname [UCATOUT=ucatl] [REUSE={NO YES}] [PWDOUT=pwdout]</pre> |

Parameters

ddname Refers to a DD or TSO ALLOC statement. The default is VSAMIN for input, and VSAMOUT for output.

dlbl Refers to a DLBL statement. The default is VDSKIN for input, and VDSKOUT for output.

dsname Name of a VSAM data set.

inds Specification of the VSAM input data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for VSAMIN is provided, no input is required.
- If no DD statement is provided specify: *dsname*.

VSE:

- If a DLBL statement for VDSKIN is provided, no input is required.
- If no DLBL statement is provided specify: *dsname*, [*ucatl*].

CMS:

- Specify: *dsname*, [*ucatl*].

key A key for KSDS records, or a slot number for RRDS records. The maximum key length is 30 characters. The first record with a key or slot value greater than or equal to *key* is the first record copied. If you omit the *key* and *skip* values, copying begins with the first record in the data set.

If the key contains lowercase characters, blanks, or commas, enclose it in single quotation marks. You can also specify a key in hexadecimal format, for example: X'C1C2C3'.

| | |
|---------------|--|
| <i>nlrecs</i> | Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>outds</i> | Specification of the VSAM output data set. (Command Syntax only.) MVS: <ul style="list-style-type: none"> • If a DD or TSO ALLOC statement for VSAMOUT is provided, no input is required. • If no DD statement is provided specify: <i>dsname</i>. VSE: <ul style="list-style-type: none"> • If a DLBL statement for VDSKOUT is provided, no input is required. • If no DLBL statement is provided specify: <i>dsname,[ucatl]</i>. CMS: <ul style="list-style-type: none"> • Specify: <i>dsname,[ucatl]</i>. |
| <i>pwdin</i> | Password of a password-protected VSAM data set. |
| <i>pwdout</i> | Password of a password-protected VSAM data set. |
| <i>reuse</i> | Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO. |
| <i>skip</i> | Number of logical records to be skipped from the beginning of the data set. If you omit the <i>skip</i> and <i>key</i> values, copying begins with the first record in the data set. |
| <i>ucat</i> | The data set name of the user catalog. |

Batch Example (MVS)

```

//VV JOB (acct),'name' VSAM to VSAM
//DITTOEX1 EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//*
/** EXAMPLE 1:
/** Copy all records from the input file to
/** the output file, starting at record 201
/**
/** Replace input.vsam.file with
/** the required input file name
/** Replace output.vsam.file with
/** the required output file name
/**
//VSAMIN DD DSN=input.vsam.file,DISP=SHR
//VSAMOUT DD DSN=output.vsam.file,DISP=SHR
/**
//SYSIN DD *
$$DITTO VV INPUT=VSAMIN,POSITION=200,
$$DITTO OUTPUT=VSAMOUT,REUSE=YES
$$DITTO EOJ
/*
//DITTOEX2 EXEC PGM=DITTO
//SYSPRINT DD SYSOUT=*
//*
/** EXAMPLE 2:
/** Copy 45 records from the input file to the
/** output file, starting at record 21
/**
/** Replace input.vsam.file with
/** the required input file name
/** Replace output.vsam.file with
/** the required output file name
/**
//SYSIN DD *
$$DITTO VV DSNIN=input.vsam.file,POSITION=20,
$$DITTO DSNOUT=output.vsam.file,REUSE=YES,
$$DITTO NLRECS=45
$$DITTO EOJ
//

```

Batch Example (VSE)

```

// JOB VV - VSAM to VSAM
// UPSI 1
// EXEC IDCAMS,SIZE=AUTO

DELETE DITTO.TC.VSAM.BV2K37 CLUSTER PURGE -
ERASE -
CATALOG(VSESP.USER.CATALOG)
DEFINE CLUSTER ( -
NAME( DITTO.TC.VSAM.BV2K37 ) -
CISZ(4096) -
FREESPACE(30 30) -
NONINDEXED -
RECORDS(1000 100) -
RECORDSIZE(200 200) -
REUSE -
VOL(SYSWK1) ) -
DATA( NAME(DITTO.TC.VSAM.BV2K37.DATA) ) -
CATALOG(VSESP.USER.CATALOG)

DELETE DITTO.TC.VSAM.BV2037 CLUSTER PURGE -
ERASE -
CATALOG(VSESP.USER.CATALOG)
DEFINE CLUSTER ( -
NAME( DITTO.TC.VSAM.BV2037 ) -
CISZ(4096) -
FREESPACE(30 30) -
NONINDEXED -
RECORDS(1000 100) -
RECORDSIZE(200 200) -
REUSE -
VOL(SYSWK1) ) -
DATA( NAME(DITTO.TC.VSAM.BV2037.DATA) ) -
CATALOG(VSESP.USER.CATALOG)
/*
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VSAMOUT,'DITTO.TC.VSAM.BV2K37',0,
VSAM,,CAT=MYCAT,DISP=(OLD,KEEP)

// UPSI 1
// EXEC DITTO
$$DITTO VER
$$DITTO BV FILEOUT=VSAMOUT,NLRECS=876,
$$DITTO RECSIZE=200,FILLCHAR=X'46',
$$DITTO KEYLOC=1
$$DITTO EOJ
/*
// UPSI 1
// DLBL MYCAT,'VSESP.USER.CATALOG',,VSAM
// DLBL VSAMIN,'DITTO.TC.VSAM.BV2K37',0,
VSAM,,CAT=MYCAT,DISP=(OLD,KEEP)
// DLBL VSAMOUT,'DITTO.TC.VSAM.BV2037',0,
VSAM,,CAT=MYCAT,DISP=(OLD,KEEP)

// EXEC DITTO
$$DITTO VER
$$DITTO VV FILEIN=VSAMIN,POSITION=10,
$$DITTO FILEOUT=VSAMOUT,NLRECS=500
$$DITTO EOJ
/*
/&

```


VX

| | |
|---------------|--|
| <i>nlrecs</i> | Number of records to be copied or ALL. The maximum number is 99 999 999. If you specify ALL or omit the parameter, all the remaining records are copied. |
| <i>pwdin</i> | Password of a password-protected VSAM data set. |
| <i>skip</i> | Number of logical records to be skipped from the beginning of the data set. If you omit the <i>skip</i> and <i>key</i> values, copying begins with the first record in the data set. |
| <i>stem</i> | A REXX stem variable name. The maximum length is 44 characters. |
| <i>ucat</i> | The data set name of the user catalog. |

Keyword Example (MVS)

```
| /* Copy a VSAM data set to a REXX stem */  
| /* Change input.vsam.file to the name */  
| /* of the required VSAM file */  
  
| "DITTO $VX DSNIN='input.vsam.file',",  
| "VARNAME=STEM."  
  
| /* Show stem contents */  
  
| do i = 1 to stem.0;  
|   say 'Record' i '=' stem.i;  
|   end;  
  
| exit;
```


WTM (Write Tape Mark)

Purpose Write one or more tape marks.

Command Syntax

```
▶—WTM—outtape—ntmks—▶
```

Batch Syntax

```
$$DITTO WTM ...
```

Keyword Syntax

```
DITTO $WTM ...
```

| MVS keywords | VSE keywords | CMS keywords |
|---|---|--|
| [LABEL=BLP] OUTPUT= <i>ddname</i> [DENS= <i>mm</i>] [NTMKS={ <u>1</u> <i>ntmks</i> }] | OUTPUT={ <i>unit</i> SYS <i>nnn</i> } [DENS= <i>mm</i>] [NTMKS={ <u>1</u> <i>ntmks</i> }] | OUTPUT= <i>unit</i> [DENS= <i>mm</i>] [NTMKS={ <u>1</u> <i>ntmks</i> }] |

Parameters

ddname Refers to a DD or TSO ALLOC statement.

LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

ntmks Number of tape marks to be written. The maximum is 9999; the default is 1.

outtape Specification of the output tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*[-*mm*].
- If label processing is desired specify:
ddname[-*mm*],[*tapevol*],[*unit*},{SL|AL},*tapedsn*,[*seqnum*],[*xdate*].
- For NL or BLP tape processing specify:
ddname[-*mm*],[*tapevol*],[*unit*},{NL|BLP},[*seqnum*],[*xdate*].

VSE or CMS:

- Specify: *unit*[-*mm*].

seqnum Relative position of the data set on the tape.

SYS*nnn* Programmer logical unit to which the tape is assigned.

tapedsn Name of the tape data set that you want to work with, if label processing is used.

tapevol Serial number (external label or V0L1) of the tape volume to be mounted, or blank for a non-specific volume request. For a volume set, specify up to five volsers in the form (volser1, volser2...).

WTM

unit MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name.
VSE or CMS: Device number (3 or 4 hexadecimal digits).

xdate Expiration date in the form *[yy]yy.ddd*.

XL (REXX Variable to Library Member) - VSE only

Purpose Copy a REXX stem variable to a library member. Each variable is copied, starting with *stem.1* and finishing with *stem.nnn*, where *nnn* is the value of *stem.0*.

For example, if *stem* is VARXX. and VARXX.0 has the value 3, the variables VARXX.1, VARXX.2, and VARXX.3 are copied to a library member.

If the library member is stored in string format, the variables are concatenated together.

This function is available only from a REXX procedure.

Command Syntax



Keyword Syntax

DITTO \$XL ...

| MVS keywords | VSE keywords | CMS keywords |
|--------------|--|--------------|
| | VARNAME= <i>stem</i> LIBOUT= <i>lib</i> MEMBEROUT= <i>member</i> [RECFMOUT={F S}] | |

Parameters

- lib* Library name, followed by a period and the sublibrary name (*library.sublibrary*).
- libdsn* The data set name of the library. If it contains blanks or commas, enclose it in quotes.
- member* Name of the library member in the form *name.type*.
- outlib* Specification of the output library. (Command Syntax only.)
 - If a DLBL statement or standard labels are available, specify: *library.sublibrary*.
 - For a library in BAM-managed space, specify: *library.sublibrary,libdsn,volser*.
 - For a library in VSAM-managed space, specify: *library.sublibrary,libdsn,,[ucatl]*.
- recfmout* Record format for the output:
 - F** Fixed-length, 80 (the default)
 - S** String format (always used if the member type is PHASE).
- stem* A REXX stem variable name. The maximum length is 44 characters.
- ucatl* The data set name of the user catalog.
- volser* Volume serial number of the device where the library is located.

Keyword Example (VSE)

```

/* REXX - copy rexx stem to a library member */
recl.1="VTMAPPL1 VBUILD TYPE=APPL"
recl.2="POWER APPL AUTH=(ACQ)"
recl.3="DBDCICS APPL AUTH=(PASS,ACQ)"
recl.0=3
"DITTO $XL VARNAME RECS. LIBOUT PRD2.CONFIG" ,
"MEMBEROUT VTMAPPL1.B"
Exit rc

```

XT (REXX Variable to Tape)

Purpose Copy a REXX stem variable to a tape file. Each variable is copied, starting with *stem.1* and finishing with *stem.nnn*, where *nnn* is the value of *stem.0*.

For example, if *stem* is VARXX. and VARXX.0 has the value 3, the variables VARXX.1, VARXX.2, and VARXX.3 are copied to a tape file.

This function is available only from a REXX procedure.

Command Syntax (MVS only)

```
▶ XT stem outtape [recfmout] [blksize] ▶
```

Command Syntax (VSE and CMS only)

```
▶ XT stem outtape [tapefile] [recfmout] [blksize] ▶
```

Keyword Syntax

DITTO \$XT ...

| MVS keywords | VSE keywords | CMS keywords |
|--|---|--|
| VARNAME= <i>stem</i> [LABEL=BLP] OUTPUT= <i>ddname</i> [DENS= <i>mm</i>] [RECFMOUT={ <u>U</u> F FB V VB VBS VS D DB DBS DS}] [BLKSIZE= <i>blksize</i>] | VARNAME= <i>stem</i> OUTPUT={ <i>unit</i> SYS <i>nnn</i> } [DENS= <i>mm</i>] [{ FILEOUT={TAPEOUT <i>tlbl</i> } OUTFILE= <i>tapefile</i> }] [RECFMOUT={ <u>U</u> F FB V VB VBS VS D DB DBS DS}] [BLKSIZE= <i>blksize</i>] | VARNAME= <i>stem</i> OUTPUT= <i>unit</i> [DENS= <i>mm</i>] [OUTFILE= <i>tapefile</i>] [RECFMOUT={ <u>U</u> F FB V VB VBS VS D DB DBS DS}] [BLKSIZE= <i>blksize</i>] |

Parameters

blksize If *recfmout* contains F, the actual block size; otherwise, the maximum block size. If *recfmout* contains B or S, *blksize* is required; otherwise, it is optional. The maximum is 65 535 (for V), 9 999 (for D), or 9 999 999 (otherwise). If the tape will be processed by other utilities or standard access methods, you should also consider the operating system limits.

ddname Refers to a DD or TSO ALLOC statement.

LABEL=BLP Specifies that bypass label processing will be used. This parameter must be specified with the first DITTO function that uses the tape. For BLP processing requirements, see “Customizing the Security Environment” in the *DITTO/ESA Installation and Customization Guide*.

mm A 2-byte tape mode code, as shown in “Tape Density and Mode Values” on page 67.

outtape Specification of the output tape unit. (Command Syntax only.)

MVS:

- If the tape is already allocated specify: *ddname*[-*mm*].
- If label processing is desired specify:
ddname[-*mm*],[*tapevol*],[*unit*,{SL|AL}],[*tapedsn*],[*seqnum*],[*xdate*].
- For NL or BLP tape processing specify:
ddname[-*mm*],[*tapevol*],[*unit*,{NL|BLP}],[*seqnum*],[*xdate*].

VSE or CMS:

- Specify: *unit[-mm]*.

| | |
|-----------------|---|
| <i>recfmout</i> | Record format for the output. The possible values are shown in the syntax diagram. Each value is a combination of the following letters: U Undefined length F Fixed length V Variable length D Variable-length ISO/ANSI tape records B Blocked S Spanned format |
| <i>seqnum</i> | Relative position of the data set on the tape. |
| <i>stem</i> | A REXX stem variable name. The maximum length is 44 characters. |
| <i>SYSnnn</i> | Programmer logical unit to which the tape is assigned. |
| <i>tapedsn</i> | Name of the tape data set that you want to work with, if label processing is used. |
| <i>tapefile</i> | VSE: the file ID of a tape file, if you want to perform label processing without using JCL. If no tape file information is available, NL processing is performed. CMS: the file ID of a tape file, if you want to perform label processing. By default, NL processing is performed. |
| <i>tapevol</i> | Serial number (external label or VOL1) of the tape volume to be mounted, or blank for a non-specific volume request. For a volume set, specify up to five volsers in the form (volser1, volser2...). |
| <i>tlbl</i> | Refers to a TLBL statement to process a labeled tape. The default is TAPEOUT. |
| <i>unit</i> | MVS: A 3 hexadecimal digit device number, a 4 hexadecimal digit device number prefixed by a slash (/), a device type, or an installation-defined group name. VSE or CMS: Device number (3 or 4 hexadecimal digits). |
| <i>xdate</i> | Expiration date in the form <i>[yy]yy.ddd</i> . |

XV (REXX Variable to VSAM)

Purpose Copy a REXX stem variable to a VSAM data set. Each variable is copied, starting with *stem.1* and finishing with *stem.nnn*, where *nnn* is the value of *stem.0*.

For example, if *stem* is VARXX. and VARXX.0 has the value 3, the variables VARXX.1, VARXX.2, and VARXX.3 are copied to a VSAM data set.

This procedure is available only from a REXX procedure.

CMS: CMS/VSAM is required.

Command Syntax

```

▶▶ XV stem outds [reuse]

```

Keyword Syntax

DITTO \$XV ...

| MVS keywords | VSE keywords | CMS keywords |
|--|---|--|
| VARNAME=stem [{ OUTPUT={VSAMOUT ddname} DSNOUT=dsname }] [REUSE={NO YES}] | VARNAME=stem [{ FILEOUT={VDSKOUT dlbl} DSNOUT=dsname [UCATOUT=ucatl] }] [REUSE={NO YES}] [PWDOUT=pwdout] | VARNAME=stem DSNOUT=dsname [UCATOUT=ucatl] [REUSE={NO YES}] [PWDOUT=pwdout] |

Parameters

ddname Refers to a DD or TSO ALLOC statement. The default is VSAMOUT.

dlbl Refers to a DLBL statement. The default is VDSKOUT.

dsname Name of a VSAM data set.

outds Specification of the VSAM output data set. (Command Syntax only.)

MVS:

- If a DD or TSO ALLOC statement for VSAMOUT is provided, no input is required.
- If no DD statement is provided specify: *dsname*.

VSE:

- If a DLBL statement for VDSKOUT is provided, no input is required.
- If no DLBL statement is provided specify: *dsname,[ucatl]*.

CMS:

- Specify: *dsname,[ucatl]*.

pwdout Password of a password-protected VSAM data set.

reuse Specify YES to replace any existing records in the output data set (if it is defined with the REUSE parameter in the catalog). Specify NO to add new records without deleting the old records. The default is NO.

stem A REXX stem variable name. The maximum length is 44 characters.

ucatl The data set name of the user catalog.

XXX (Function Summary)

Purpose Print a summary of DITTO functions and parameters.

Use this function to obtain a quick reference guide of the DITTO functions that are available to you.

Command Syntax

▶—XXX—◀

Batch Syntax

\$\$DITTO XXX

Keyword Syntax

DITTO \$XXX

Parameters: None.

XXX

Part 4. Appendixes

For users new to DITTO

Part 1, "Introduction to DITTO" on page 1 gives an overview of DITTO functions, and the different ways you can use DITTO.

For users already familiar with DITTO basics

Part 2, "Guide to Using DITTO Functions" on page 29 describes what you can accomplish with DITTO functions. The first two chapters describe browse and edit functions that are available in full-screen mode. The remaining chapters are divided according to the type of input you are working with: catalog entries, tapes, disk records, sequential data (QSAM or SAM), VSAM data, CMS files, VSE library members, OAM objects, or cards.

For experienced DITTO users

Part 3, "DITTO Reference" on page 147 presents the DITTO functions in alphabetical order.

For special situations

Part 4, "Appendixes" contains information on compatibility with previous releases, installing DITTO, controlling DITTO security, and responding to DITTO error messages.

If you are running DITTO in full-screen mode, you can also refer to the online help.

Appendix A. Compatibility with Previous Releases of DITTO

This appendix describes the differences between DITTO/ESA Release 3 and previous versions of DITTO:

- MVS/DITTO Version 2 Release 1 (Program Number 5695-100)
- DITTO Version 3 Release 2 for VSE and VM (Program Number 5688-052)
- DITTO Version 3 Release 2 Productivity Features for VSE and VM

Under VSE and CMS, the DVT and VDL functions have been changed. Previously, an asterisk (*) within a file ID represented any number of characters. Now, one asterisk represents any number of characters within a qualifier and a double asterisk (**) represents any number of characters in any number of qualifiers.

Under VSE and CMS, an implicit rewind was previously performed by each function that works with labeled tapes. Labeled tape functions could only work with the first file on a tape. The implicit rewind is no longer performed. This lets you work with multifile standard labeled tapes.

DITTO Functions That are No Longer Supported

The following table lists function codes that were allowed in previous releases of DITTO but are not recognized by DITTO/ESA. You can use the indicated replacement, if any.

| Function | Description | Replacement |
|----------|---------------------------------|-------------|
| BDU | Buffer to Diskette | - |
| BIS, BI | Buffer to ISAM | - |
| CD | Card Dump | CP |
| CDU | Card to Diskette | - |
| CI | Card to Card Interpreted | CC |
| DD | Disk Dump | DP |
| DDD | Disk Dump with Deblocking | DP |
| DPD | Disk to Printer with Deblocking | DP |
| DKB | Diskette Browse | - |
| DKE | Diskette Eject and Feed | - |
| DKH, DKP | Diskette to Printer | - |
| DKI | Display Diskette Index | - |
| DKL | Diskette Record Load | - |
| DKN | Diskette to Console | - |
| DKS | Diskette Record Scan | - |
| DKV | Diskette Identification Change | - |
| DUC, DUI | Diskette File to Card | - |
| DUD | Diskette File to Diskette File | - |
| DUF | Diskette File to CMS File | - |

Appendixes

| Function | Description | Replacement |
|----------|-------------------------------------|-------------|
| DUH, DUP | Diskette File to Printer | - |
| DUS | Diskette File to SAM File | - |
| DUT | Diskette File to Tape | - |
| DUV | Diskette File to VSAM | - |
| FD | CMS File Dump | FP |
| FDD | CMS File Dump Deblocked | FP |
| FDU | CMS File to Diskette File | - |
| IDK | Initialize Diskette | - |
| IDP, ID | ISAM File Dump | - |
| IIS | ISAM File to ISAM File | - |
| IPR, IP | ISAM File Print | - |
| ISQ, IS | ISAM File to SAM File | - |
| ITP, IT | ISAM File to Tape | - |
| IVS, IV | ISAM File to VSAM | - |
| LD | Library Member Dump | LP |
| LE | Library Member Erase | LDEL |
| LI | Library Member to Card Interpreted | LC |
| OD | Object Dump | OP |
| QD, SDP | Sequential Data Dump | SP |
| QI, SI | Sequential Data to Card Interpreted | SC |
| SD, SDD | Split-Cylinder Disk Dump | - |
| SDU | SAM to Diskette | - |
| SIS | SAM File to ISAM File | - |
| SP, SPD | Split-Cylinder Disk Print | - |
| SRS | Split-Cylinder Disk Record Scan | - |
| TD | Tape Dump | TP |
| TDD | Tape Dump Deblocked | TP |
| TDU | Tape to Diskette | - |
| TDV | Tape Dump Variable | TP |
| TI | Tape to Card Interpreted | TC |
| TIS | Tape to ISAM File | - |
| TPD | Tape Print Deblocked | TP |
| TPV | Tape Print Variable | TP |
| VD, VDP | VSAM Dump | VP |
| VDU | VSAM to Diskette File | - |
| VI | VSAM to Card Interpreted | VC |
| VIS | VSAM to ISAM File | - |

DITTO Function Code Synonyms

The following table lists supported synonyms for DITTO function codes.

| Function | Synonym(s) | Description |
|----------|------------|------------------------------------|
| BS | BQ, BSQ | Create Sequential Data |
| BT | BTP | Create Tape File |
| BV | BVS | Create VSAM File |
| CS | CQ, CSQ | Card to Sequential Data |
| CV | CVS | Card to VSAM |
| DUMP | DUM | Dump CMS File to Tape |
| FP | FPR | CMS File Print |
| FT | FTP | CMS File to Tape |
| FV | FVS | CMS File to VSAM |
| LOAD | LOA | Load CMS File from Tape |
| MB | TST | Memory Browse |
| NEW | NEWS | News Command |
| OS | OQ, OTL | Object to Sequential Data |
| SC | QC | Sequential Data to Card |
| SO | QO, TLO | Sequential Data to Object |
| SP | QP, SPR | Sequential Data Print |
| SS | QQ, SSQ | Sequential Data to Sequential Data |
| ST | QT, STP | Sequential Data to Tape |
| SV | QV, SVS | Sequential Data to VSAM |
| TS | TQ, TSQ | Tape to Sequential Data |
| TV | TVS | Tape to VSAM |
| VP | VPR | VSAM Print |
| VRU | VRL | VSAM Record Update |
| VS | VQ, VSQ | VSAM to Sequential Data |
| VT | VTP | VSAM to Tape |
| VV | VVS | VSAM to VSAM |

Batch Keywords That are No Longer Supported

The following table lists keywords that were allowed in previous releases of DITTO but are not recognized by DITTO/ESA. You can use the indicated replacement, if any.

| Function(s) | Keyword | Description | Replacement |
|-------------|------------|-------------|------------------------------|
| BV | BLKFACTOR= | - | Output blocking factor |
| TMP | TYPE=CHAR | FORMAT=CHAR | Tape map in character format |

Batch Keywords That are Not Recommended

The following table lists obsolete keywords from previous releases of DITTO that are still recognized by DITTO/ESA in batch mode. It is recommended that you use the indicated replacement, if any.

| Function(s) | Keyword | Description | Replacement |
|--|-------------------------|-------------------------------------|-----------------------|
| BS, BT, CS, CT, FT, SS, ST, TFT, TS, TTR, VS, VT | BLKFACTOR= | Output blocking factor | RECFMOUT= BLKSIZE= |
| TF | BLKFACTOR= | Output blocking factor | - |
| BS, CS, SS, TS, VS | CISIZE= | FBA control interval size | BLKSIZE= |
| FT | MODE= | Input Record format | - |
| SP (VSE, CMS), SS, ST (VSE, CMS), SV, TC, TF, TFT, TRS, TS, TTR, TV | MODE= | Input Record format | RECFMIN= |
| BT, CT, ST (MVS), VS, VT | MODE= | Output record format | RECFMOUT= |
| TTR | OUTMODE= | Output record format | RECFMOUT= |
| TP, TRS | NBLKS= | Number of blocks | NLRECS= |
| FT | RECSIZE= | Record length for deblocking | - |
| VC, VL, VP, VRU, VS, VT, VV | START=K POSITION=key | VSAM start positioning by key value | KEY=key |
| SET | TSOPRINT= | Print output destination | PRINTOUT= |
| SET | ASCII=YES | Translate from ASCII to EBCDIC | ASCII=IN |
| <p>Notes:</p> <ol style="list-style-type: none"> 1. For the SP function, the MODE keyword is obsolete under VSE and CMS but still applies under MVS. 2. For the ST function, the MODE keyword is obsolete. Under VSE and CMS, use RECFMIN; under MVS, use RECFMOUT. | | | |

Appendix B. DITTO Messages

DITTO issues messages to:

- Confirm successful processing
- Prompt you for more information
- Tell you about errors

DITTO messages that need more explanation than can be included in a few words are preceded by DIT and a message number. These numbered messages are described here in order by message number.

In full-screen mode, numbered messages are displayed either just above or just below the command line depending on where the command line is positioned. You can get more information about a numbered message by pressing the Help function key.

Format of Numbered Messages

The format of numbered DITTO messages is:

DITTO Message Syntax

DITnnnx message text

DIT Indicates that the message comes from DITTO.

nnnn Is the message number.

x Is the action indicator:

A Action. The operator must take action, for example, ready a device.

D Decision. This message requires a response.

I Information. This message does not require a response, but informs you that an error has occurred.

DIT0090I Insufficient virtual storage available

Explanation: DITTO does not have enough virtual storage to start or to continue the current function.

User Response: MVS: Run DITTO with a larger region size.

VSE: Rerun the job in a larger partition. Invoke full-screen DITTO with the JOBCLASS parameter and specify the class of a larger partition.

CMS: Define a larger virtual storage size for your virtual machine, re-IPL CMS, and restart DITTO.

DIT0110I Function *name* is not supported in batch mode

Explanation: You called a DITTO function that cannot be run in batch mode.

User Response: Use a different function, or invoke DITTO in a different mode.

DIT0120I Function *name* is not included in DITTO/ESA

Explanation: You called a function that is not part of DITTO/ESA.

User Response: Refer to Appendix A, "Compatibility with Previous Releases of DITTO" on page 349 for details of functions that have changed with DITTO/ESA.

DIT0130I Function *name* is not supported in full-screen mode

Explanation: You called a DITTO function that cannot be run in full-screen mode.

User Response: Use a different function, or invoke DITTO in a different mode.

DIT0140I Function *name* is not supported in command mode

Explanation: You called a DITTO function that cannot be run in command mode.

User Response: Use a different function, or invoke DITTO in a different mode.

DIT0150I Function *name* is not supported in line mode

Explanation: You called a DITTO function that cannot be run in line mode.

User Response: Use a different function, or invoke DITTO in a different mode.

DIT0161I Function *name* is not supported in MVS

Explanation: You called a DITTO function that is not available under the MVS operating system.

User Response: For a list of functions, refer to Chapter 1, "Overview of DITTO Functions" on page 3, or to the list produced by the XXX function.

DIT0162I Function *name* is not supported in CMS

Explanation: You called a DITTO function that is not available under CMS.

User Response: For a list of functions, refer to Chapter 1, "Overview of DITTO Functions" on page 3, or to the list produced by the XXX function.

DIT0163I Function *name* is not supported in VSE

Explanation: You called a DITTO function that is not available under the VSE operating system.

User Response: For a list of functions, refer to Chapter 1, "Overview of DITTO Functions" on page 3, or to the list produced by the XXX function.

DIT0170I Function *name* is not supported in stand-alone VSE

Explanation: You called a DITTO function that is not available in a stand-alone VSE environment.

User Response: Use a function that is available in the VSE stand-alone environment. These functions are listed in "VSE: Functions Supported in Stand-Alone Environment" on page 27 and in the list produced by the XXX function.

DIT0200I Function *xxxx* only supported in a CMS/VSE DITTO session

Explanation: You called a function that is not supported when DITTO is started from a CICS terminal.

User Response: Start VSE/DITTO from a CMS session. Refer to "CMS/VSE Mode: Starting VSE/DITTO from CMS" on page 14.

DIT0300I DITTO security setup failed - RPIUCMS R15 X'*nn*'

Explanation: DITTO is unable to establish the connection to RACF. For more information, refer to the RPIUCMS message and return code.

User Response: Ensure that your RACF environment is set up correctly. Refer to your RACF documentation and to the *DITTO/ESA Program Directory*.

DIT0310I DITTO security setup failed - RACROUTE R15 X'nn', RC X'nn', RSCD X'nn',

Explanation: DITTO is unable to get RACF information via RACROUTE. For more information, refer to the RACROUTE return and reason codes.

User Response: Ensure that your RACF environment is set up correctly. Refer to your RACF documentation, the *DITTO/ESA Program Directory*, and the *DITTO/ESA Installation and Customization Guide*.

DIT0321I DITTO security setup failed - module 'DITSECUR' not found in LPA

Explanation: The security exit must exist in LPA.

User Response: Refer to the *DITTO/ESA Program Directory* and the *DITTO/ESA Installation and Customization Guide*.

DIT0322I DITTO security setup failed - module 'DITSECUR' not found

Explanation: The security exit must exist.

User Response: Refer to the *DITTO/ESA Program Directory* and the *DITTO/ESA Installation and Customization Guide*.

DIT0330I Not authorized for function

Explanation: Your user ID is not authorized to use the function that you specified.

User Response: If you need to use this function, contact your system support to enable your user ID for this function or function group. Refer to the *DITTO/ESA Installation and Customization Guide*.

DIT0350I Not authorized to xxx spool output not owned

Explanation: Your user ID is not authorized to access the specified spool file.

User Response: If you need to use this function, contact your system support to enable your user ID to access spool output not owned by you. Refer to the *DITTO/ESA Installation and Customization Guide*.

DIT0360I Not authorized for fullpack access to VOLSER volser

Explanation: Your user ID is not authorized to use the fullpack disk functions on the specified volser.

User Response: If you need to use fullpack disk functions, contact your system support to enable your user ID for fullpack access as described under "Customizing the Security Environment" in the *DITTO/ESA Installation and Customization Guide*.

DIT0370I Function name not supported in authorized mode

Explanation: You invoked a function that cannot be run in authorized mode.

User Response: Invoke DITTO in unauthorized mode.

DIT0400I Panel display error rc for panel name

Explanation: The panel cannot be displayed (for example, the panel may have been changed for translation).

User Response: Save any system error information and contact your system support.

DIT0410I Unable to initialize full-screen mode

Explanation: DITTO was unable to set up a full-screen session, because an unexpected error occurred.

User Response: Save any system error information and contact your system support.

DIT0490I Invalid value for parameter parameter

Explanation: You specified an incorrect value for a parameter in the DITTO/CICS invocation.

User Response: Refer to "Running DITTO in Full-Screen Mode" on page 11.

DIT0500I Invalid command syntax, 'parm' is unknown

Explanation: The DITTO CICS transaction was invoked with invalid syntax.

User Response: Refer to "Running DITTO in Full-Screen Mode" on page 11.

DIT0510I CICS storage request failed

Explanation: The CICS partition does not have enough free virtual storage to start DITTO. Too many transactions are currently active.

User Response: Try to invoke DITTO again later.

**DIT0520A Enter job userid and password
==> userid ==> password**

Explanation: VSE security is active but you did not specify a password. A user ID and password are needed to access secured resources.

User Response: Specify a user ID and password if you want to access secured resources. You can specify the user ID and password directly when invoking the DITTO transaction, as described "Running DITTO in Full-Screen Mode" on page 11.

DIT0530I No response, see job 'jobname' output and log

Explanation: DITTO did not start successfully in the batch partition.

User Response: Check the console output and the job output for error conditions.

DIT0540I Terminated abnormally, see job 'jobname' output and log

Explanation: DITTO terminated abnormally.

User Response: Check the console output and the job output for error conditions.

DIT0550I Jobclass 'jobclass' not active or in use

Explanation: DITTO did not start successfully in the batch partition. No partition for the specified job class is available.

User Response: Try the command again. If possible, specify a different job class. Refer to “Running DITTO in Full-Screen Mode” on page 11 for the command syntax.

DIT0560I Terminated due to an internal error

Explanation: DITTO encountered an internal problem.

User Response: Contact your system support.

DIT0570I system communication error, REQUEST request RC rc RS rsn

Explanation: A communication error occurred between the CICS transaction and the batch partition or VSE/POWER.

User Response: Contact your system support.

DIT0580I Communication error, POWER does not respond

Explanation: The CICS transaction cannot communicate with VSE/POWER.

User Response: Contact your system support.

DIT0590I Module 'DITJOBFS' is inactive or missing

Explanation: The DITTO transaction could not load the module DITJOBFS. This module contains the job skeleton to start DITTO.

User Response: Check with the CEMT transaction if the program DITJOBFS is defined and enabled. For more information, refer to the *DITTO/ESA Installation and Customization Guide*.

DIT0600I DITTO VSE connection rejected

Explanation: The DITTO CMS display server could not be started. The server is not defined to IUCV correctly.

User Response: Ensure that the correct server name is specified in the following CP directory entry for the failing user:

```
IUCV *IDENT server-name GLOBAL
```

The server name is built by the DITTOVSE EXEC and can be customized. The default is the user ID prefixed with the letter S and truncated to 8 characters.

DIT0610I DITTO VSE disconnected abnormally

Explanation: The communication to the DITTO VSE batch partition terminated abnormally.

User Response: Contact your system support. Supply the DITTO VSE job and console log.

DIT0620I Unexpected return code, RC nnn RSN rrrr from DITTO display server

Explanation: The DITTO CMS display server returned an unexpected return code.

User Response: This is probably a DITTO error. Contact your system support.

DIT0630I Not authorized to connect to *IDENT

Explanation: The DITTO CMS display server could not be started. The server is not defined to IUCV correctly.

User Response: Check that the CP directory entry for the user contains the following entries:

```
OPTION MAXCONN 2 (or higher)
IUCV ALLOW
IUCV *IDENT RESANY GLOBAL
```

or

```
IUCV *IDENT server-name GLOBAL
```

The server name is built by the DITTOVSE EXEC and can be customized. The default is the user ID prefixed with the letter S and truncated to 8 characters.

DIT0640I No VSE user ID specified

Explanation: The name of the target VSE system has not been supplied.

User Response: Customize the DITTOVSE EXEC (specify a default VSE name) or specify the VSE option when calling the DITTOVSE EXEC.

DIT0650I DITTO profile save failed

Explanation: DITTO was updating the profile and an error occurred.

User Response: MVS: Check the DITPROF allocation or the DITPROF specification in the installation profile. Refer to the *DITTO/ESA Installation and Customization Guide*.

VSE: Check the CATALOG parameter specification in the LIBDEF statement of your DITTO startup job. Refer to the *DITTO/ESA Installation and Customization Guide*.

CMS: provide an A-disk in read/write mode.

DIT0660I DITTO profile not found or in error, defaults used

Explanation: The DITTO installation or user profile was not found or is in error. The default values supplied by IBM are used.

User Response: Refer to the *DITTO/ESA Installation and Customization Guide*.

DIT0670I DITTO profile parameter *parameter=value* unknown, defaults used

Explanation: An invalid parameter was found in the DITTO installation profile. The default values supplied by IBM are used.

User Response: Refer to the *DITTO/ESA Installation and Customization Guide*.

DIT0680I Profile error on or near '*parameter*', DITTO defaults used

Explanation: A syntax error was detected in the DITTO installation profile, in or near the indicated parameter. The job continues using the IBM-supplied defaults.

User Response: Correct the user-supplied DITTO profile. Refer to the *DITTO/ESA Installation and Customization Guide*.

DIT0690I Panel library problem, DITPLIB DD and *dsname* not found

Explanation: DITTO failed to access the panel library. DITTO continues in line mode.

User Response: You may have changed the default installation name of the panel library. Check your DITPLIB DD statement in your logon procedure or the DITPLIB parameter specification in the installation profile. Refer to the *DITTO/ESA Installation and Customization Guide*.

DIT0800I SYSLST is assigned to disk, output truncated to 121 bytes

Explanation: VSE SYSLST data sets on disk have a length restriction of 121 bytes. Because the output length in DITTO is 133 bytes, writing to SYSLST causes the output to be truncated. Processing continues.

DIT0810I Print record not in forms 'A' format

Explanation: A TFA or SFA function found an invalid printer CCW code in the first byte of a tape or disk record.

User Response: Make sure that the input data to be used is in SFA/TFA format. If the input is type D, use the SFD or TFD function, instead.

DIT0830I RECLIMIT *nnnnn* start value exceeds *nnnnn* record length

Explanation: The SET parameter RECLIMIT specifies a start position that is greater than the record length of the record you are trying to print.

User Response: Use the SET function and set the RECLIMIT start to a position within the record, then rerun the function.

DIT0841I REXX not available, printout routed to SYSPRINT

Explanation: You specified SET PRINTOUT=REXX, but DITTO was not called from a REXX procedure. The print output will be routed to SYSPRINT instead. (SET PRINTOUT=SYSPRINT).

User Response: Use SET PRINTOUT=REXX only when calling DITTO from a REXX procedure.

DIT0842I REXX not available, printout routed to printer

Explanation: You specified SET PRINTOUT=REXX, but DITTO was not called from a REXX procedure. The print output will be routed to the printer instead. (SET PRINTOUT=PRINTER).

User Response: Use SET PRINTOUT=REXX only when calling DITTO from a REXX procedure.

DIT0910I Missing or invalid control card

Explanation: The message indicates an error in a control card. A subsequent message contains more information about the error.

User Response: Correct the control card and rerun the job.

DIT0920I Parameter *parameter* is missing

Explanation: You omitted a required parameter.

User Response: Provide all required parameters.

DIT0930I Missing or invalid function code

Explanation: You either omitted a function code or specified an invalid function code.

User Response: Provide the correct function code in the control statement.

DIT0940I Invalid syntax near card column *nn*

Explanation: The syntax of the control statement is invalid. A scale is printed to help you find the error.

User Response: Correct the control statement.

DIT0950I Too many parameters

Explanation: You may have specified excessive or duplicate parameters.

User Response: Remove excessive or duplicate parameters.

DIT0960I Parameters *parameter1* and *parameter2* are mutually exclusive

Explanation: You specified two parameters that cannot be used together.

User Response: Provide the correct parameters.

DIT0970I Parameter *parameter1* or *parameter2* missing

Explanation: You did not supply a required parameter.

User Response: Specify one of the indicated parameters.

DIT0980I Parameter *parameter1* or *parameter2* or *parameter3* missing

Explanation: You did not supply a required parameter.

User Response: Specify one of the indicated parameters.

DIT0990I Invalid continuation, syntax error near card column *nn*

Explanation: An expected continuation of the control statement could not be found or contains invalid syntax.

User Response: Either remove the continuation indicator or correct the continuation card.

DIT1000I Unexpected end of parameter specification

Explanation: The parameters for the DITTO invocation are terminated by a comma.

User Response: Correct the parameter specification for DITTO invocation.

DIT1130I Parameter *parameter* invalid or not applicable for this function

Explanation: You specified an unknown parameter or a parameter that is not applicable for this function.

User Response: Correct the DITTO invocation or the control statement.

DIT1140I Parameter *parameter* is ambiguous

Explanation: DITTO cannot determine an abbreviated parameter that you specified.

User Response: Specify the full parameter name.

DIT1150I Value of parameter *parameter* missing

Explanation: A value for the specified parameter could not be found.

User Response: Correct the DITTO invocation or the control statement.

DIT1160I Extraneous parameter *parameter* ignored

Explanation: You supplied a parameter that is not used by this function, or a duplicate parameter.

User Response: Remove the parameter from the DITTO invocation or the control statement.

DIT1170I Invalid value for parameter *parameter*

Explanation: You specified an incorrect value for a parameter.

User Response: Correct the DITTO invocation or the control statement.

DIT1180I Value of parameter *parameter* too long

Explanation: You specified an incorrect value for a parameter.

User Response: Correct the DITTO invocation or the control statement.

**DIT1190I Value out of range for parameter
parameter**

Explanation: You specified an incorrect value for a parameter.

User Response: Correct the DITTO invocation or the control statement.

**DIT1200I Invalid or inconsistent KEYLOC or
KEYLEN value**

Explanation: The key length value and key location value that you specified would result in a key that does not fit into the record where it is to be used.

The key location, plus the key length, minus 1 must be less than or equal to the record length. (For example, if the key location is 50 and the key length is 10, the record length must be 59 or more.)

User Response: Change the key length, the key location, or both.

DIT1210I Increment value too high

Explanation: The increment value does not fit within the key length specified.

User Response: Change increment or key length.

DIT1301I No print feature on SYSPUNCH device

Explanation: You called a function that interprets punched cards, but your card punch cannot interpret cards.

User Response: Copy to cards without interpreting, or allocate SYSPUNCH to a card punch that has a print feature.

DIT1302I Device *unit* not capable or spooled

Explanation: You called a function that interprets punched cards, but your card punch cannot interpret cards, or the device is not dedicated.

User Response: Copy to cards without interpreting, or attach/assign a punch device that has a print feature.

**DIT1330I *ddname* DD is allocated to a dummy
device**

Explanation: There is no data set available for processing. DITTO cannot process dummy data sets.

User Response: Allocate a real data set, and rerun the function.

**DIT1340I DITIIPRT DD must be same unit as
SYSPUNCH DD**

Explanation: The II (Interpret Cards) function reads input from SYSPUNCH and writes output to DITIIPRT.

User Response: Allocate DITIIPRT with unit affinity to SYSPUNCH.

DIT1500I End of *xxxxx* tape sensed

Explanation: The indicated tape is positioned at the end of the tape (that is, after the end-of-tape (EOT) mark). If the tape is a reel, it might be pulled off the feeding reel.

User Response: If the tape stops before its physical end, you may proceed with caution if required (for example to copy a broken tape to its physical end).

**DIT1560I End of *xxxxx* tape sensed, function
terminated**

Explanation: The indicated tape is positioned at the end of the tape (that is, after the end-of-tape (EOT) mark). If the tape is a reel, it might be pulled off the feeding reel by subsequent forward processing of the tape.

DIT1600I Unit *unit* does not support Erase Tape

Explanation: You called the Erase Tape function, but the tape unit cannot perform the Erase Tape function.

User Response: Mount the tape on an IBM 3400 tape unit (or its equivalent) to perform this function.

**DIT1621I Error on *xxxxx* tape, ECB *ecb*, CSW *csw*,
sense 0-3 *sense***

Explanation: An error occurred on the indicated tape unit. If a read data check occurred, message DIT1630D may follow.

User Response: Save the message text. Scan the console log for an I/O error message, and save it. If the error persists, give the message text to your system support.

**DIT1622I Error on *xxxxx* tape, DIAG*xx*, CC *cc*, RC
rc, CSW *csw*, sense 0-3 *sense***

Explanation: An error occurred on the indicated tape unit. If a read data check occurred, message DIT1630D may follow.

User Response: Save the message text. Scan the console log for an I/O error message, and save it. If the error persists, give the message text to your system support.

DIT1623I Error on xxxxx tape, CCBCOM com, CSW csw, sense 0-3 sense

Explanation: An error occurred on the indicated tape unit. If a read data check occurred, message DIT1630D may follow.

User Response: Save the message text. Scan the console log for an I/O error message, and save it. If the error persists, give the message text to your system support.

DIT1630D Bypass record - B, ignore error - I, user correct - C

Explanation: An error was found reading from an input tape.

User Response: Reply B to skip the record and read the next record. Reply I to use the record as it was read into the input buffer. Reply C to correct the record.

DIT1700I No EOD delimiter is set

Explanation: You specified EOD as the number of files in a tape function, but you have not specified an EOD value with the SET function. There is no default EOD value for tapes.

User Response: Specify a different number of files or use the SET function to define an EOD delimiter.

DIT1710I Input record exceeds nnnnn byte buffer

Explanation: DITTO could not allocate a buffer large enough for the record.

User Response: Rerun the function with more virtual storage.

DIT1720I nnnnn byte input block exceeds nnnnn byte buffer

Explanation: DITTO could not allocate a buffer large enough for the block.

User Response: Rerun the function with more virtual storage.

DIT1810I No more data found on input tape

Explanation: The end of data on an input tape has been reached. (If a data check occurs immediately after a tape mark, DITTO assumes that the end-of-data has been reached.)

DIT2030I Software EOF found

Explanation: DITTO found a control interval of all zeros (software end-of-file) at the specified disk location. Therefore, no further processing was performed.

User Response: To continue with the FBA blocks following the software EOF, omit the CISIZE specification.

DIT2040I No control interval found at PBN nnnn

Explanation: You specified a CISIZE other than zero, but no valid control information was found. This may be caused by an incorrect CISIZE parameter value or by a BEGIN parameter value that does not denote a CI boundary.

User Response: Correct the CISIZE parameter, the BEGIN parameter, or both. For more information about the cause, omit the CISIZE parameter. You can then view the data as stored on the FBA device.

DIT2050I Track overflow occurred during write CKD

Explanation: A track overflow condition occurred while writing to a CKD disk. This occurs during the DRL function when trying to replace an EOF record with a data record. DITTO restores the end-of-file (EOF) record.

DIT2180I No disk record found

Explanation: The specified disk record does not exist on this track.

User Response: If appropriate, specify a lower disk record number.

DIT2190I No home address record

Explanation: A home address record was not found at the specified disk location. This is probably a hardware error.

User Response: Rerun the function. If the problem recurs, contact your system support.

DIT2200D Should new EOF be written after this record? Y or N

Explanation: The DRL function lets you change an end-of-file record into a record that has a KEY and DATA field (convert EOF to a data record).

User Response: If you want a new end-of-file record to follow this record, reply Y. Otherwise, enter N or U to exit the function.

DIT2210I EOF scan requires CISIZE

Explanation: EOF scanning was specified for an FBA device, but no CISIZE value was given.

User Response: Rerun the function with a CISIZE specified.

DIT2520I Output tape not positioned at load point or after EOF labels

Explanation: The output tape is not positioned correctly to write a labeled data set. Valid positions are the load point and after another labeled file.

User Response: Ensure that the tape is positioned correctly and rerun the function, or rerun the function without label processing.

DIT2530I No xxxxxx label found on input tape

Explanation: The tape indicated is not a standard label tape, or is not positioned correctly.

User Response: Mount a labeled tape. Position it at the data set you want to process and rerun the function, or rerun the function without label processing.

DIT2540I No xxxxxx label found on output tape

Explanation: The output continuation volume is not a standard label tape.

User Response: Ensure that all output volumes are initialized with standard labels and rerun the function, or rerun the function without label processing.

DIT2560A Wrong continuation volume, mount correct tape volume

Explanation: The tape mounted is not the continuation of the preceding volume.

User Response: Mount the correct continuation volume.

DIT3010I unit is invalid device for this function

Explanation: The specified device has a type which cannot be used in the function you invoked. (For example, the input device for a disk function is a tape unit.)

User Response: Use a valid device for this function, or use a DITTO function appropriate for the device specified.

DIT3030I SYSnnn assignment invalid

Explanation: The device assigned to SYSnnn is not the correct type for the function you invoked. (For example, the input device for a tape function is a disk.)

User Response: Correct the assignment or the control statement.

DIT3040I 'volser' volume information not available, GETVCE RC rc

Explanation: The disk with volume identifier *volser* is not known to the system.

User Response: Select a valid volume serial number. If the problem persists for an online volume, contact your system support.

DIT3050I SYSnnn is unassigned

Explanation: A programmer logical unit referred to in an INPUT or OUTPUT parameter, or EXTENT JCL statement, or a system logical unit used by DITTO, is not assigned.

User Response: Correct the JCL or INPUT or OUTPUT parameter, and rerun the job.

DIT3060I No LUBs available

Explanation: The programmer logical unit could not be assigned dynamically because the logical unit block (LUB) table is full.

User Response: Select a device for which an assignment already exists, or release any unnecessary assignments and rerun the function.

DIT3100I Data set dsname not found

Explanation: No data set *dsname* was found on the unit specified, or in the VSAM catalog used.

User Response: MVS: Correct the data set name in the DD statement, in the DITTO invocation, or in the control statement.

VSE: Ensure that any DLBL and EXTENT statements used contain correct data set name, volume, and catalog information, and rerun the job. In an interactive DITTO session, give the correct replies.

CMS: Ensure that the correct volumes are linked and accessed. For a VSAM data set, invoke function VMC and check the VSAM master catalog definition. Invoke the function again and give the correct replies.

DIT3111I No *ddname* DD statement supplied

Explanation: *ddname* is a *ddname* referred by an INPUT or OUTPUT parameter, or any other *ddname* used by DITTO. Either the corresponding DD statement is missing, or the *ddname* was used and freed by a TLT function.

User Response: Correct the JCL or the INPUT or OUTPUT parameter, and rerun the job. To access a tape again after a TLT function, end the job step and continue with another step.

DIT3112I DLBL statement *dlblname* missing or invalid

Explanation: The DLBL name given in a FILEIN or FILEOUT parameter could not be found in the label area, or the DLBL statement that was found cannot be used for the function you called.

User Response: Supply a DLBL and EXTENT statement for *dlblname*, or correct the name in the FILEIN or FILEOUT parameter.

DIT3120I EXTENT information for *name* missing or inconsistent

Explanation: An EXTENT statement was not supplied for the named DLBL, or the device information is missing or does not match the DITTO parameters.

User Response: Supply or correct the EXTENT information.

DIT3130I TLBL statement *tlblname* missing

Explanation: A tape function was called using a FILEIN or FILEOUT parameter, but no TLBL statement was supplied for *tlblname*.

User Response: Supply a TLBL statement.

DIT3140I WARNING: *dsname* in use. *function* continues without exclusive control

Explanation: You invoked a disk update function for a data set already in use by another task.

User Response: Ensure that your update request does not expose the data or the other task and continue, or run the function when the data set is available for exclusive control.

DIT3150I Data set request denied by SVC99 validation routine

Explanation: A dynamic allocation request was rejected by your system's SVC99 validation routine. This is probably due to a restriction imposed by your system.

User Response: Contact your system support.

DIT3161I *ddname* DD open failed, ABEND code *abend-reason*

Explanation: The data set or tape specified by *ddname* could not be opened.

User Response: Refer to the accompanying system message, and check the status of the data set or tape.

DIT3162I DLBL/TLBL open failed, cancel code *abend*

Explanation: The data set specified by the DLBL or TLBL name could not be opened.

User Response: Refer to the accompanying system message, and check the status of the data set.

DIT3170I Invalid *xxxxxx* data set for function *xxx*

Explanation: The function that you specified cannot be used with the data set that you specified. For example, a VSAM function cannot process a sequential data set; a SAM function can be used only for a sequential data set.

User Response: Specify a function capable of processing the data. You can use basic disk functions to inspect data with unknown organization.

DIT3181I *type* data set *ddname* macro failed, ABEND code *xxxx-xxx*

Explanation: The indicated operation failed for the indicated data set.

User Response: Check for any accompanying system messages, and refer to the appropriate system manual.

DIT3182I *type* data set *dlbl/tlbl* macro failed, cancel code *xxxx*

Explanation: The indicated operation failed for the indicated data set.

User Response: Check for any accompanying system messages, and refer to the appropriate system manual.

DIT3190I Unable to process CISIZE 0

Explanation: A SAM input function was invoked, and the input data set was specified to be on an FBA device. The CISIZE field in the format-1 label of the input data set contains zero, indicating that the data set was created through DTFPH. This data set cannot be processed by the data set functions of DITTO.

DIT3200I No SYSPRINT DD statement supplied

Explanation: DITTO requires SYSPRINT to be allocated.

User Response: Ensure that a SYSPRINT DD statement is supplied, then rerun the job.

**DIT3210I EOVS for xxxx failed, ABEND code
xxxx-xxx**

Explanation: An ABEND occurred during EOVS processing for the indicated tape.

User Response: Check for any accompanying system messages, and refer to the appropriate system manual. If the problem persists, contact your system support.

DIT3310I Permanent I/O error on input/output

Explanation: An unexpected return code was returned for a read or write operation.

User Response: Refer to the corresponding system message for more information.

**DIT3320I Permanent data check on input/output,
CSW csw, sense code**

Explanation: This is probably a hardware error.

User Response: Refer to the hardware device reference manual for an interpretation of the CSW and the sense bytes.

**DIT3330I Read error, incorrect block length on
input**

Explanation: A block of the input data set does not correspond to the data set's format description from the VTOC or DD statement.

User Response: Change the DCB parameter of the DD and rerun the job, or try using a tape or disk function instead of a QSAM function to process this data set.

**DIT3500I Block size (nnnnn) invalid for
FIXED,nnnnn output**

Explanation: You cannot write records in fixed, unblocked format to a data set with a block size not equal to the record size.

User Response: Specify a block size equal to the record size or a different record format.

**DIT3510I Input data length (length) not multiple of
reclen (reclen)**

Explanation: An input block or record has a length that is not equal to, or a multiple of, the output record size.

User Response: Correct the input specification or use the SET PAD= function to adjust the input records.

**DIT3520I Block size (nnnnn) exceeds maximum
(max)**

Explanation: MVS: For a QSAM output function, you specified a block size greater than the maximum supported value.

VSE or CMS: You tried to write a block that is too large for the type of data set you are working with. The block size may be limited by the record format used, the track capacity of a CKD device, the FBA control interval size, or the attributes of a data set defined in the catalog.

User Response: MVS: Specify a smaller block size.

VSE or CMS: Specify a smaller record size, or change the output data set definition.

**DIT3530I Record size (reclen) exceeds maximum
(max)**

Explanation: MVS: A record exceeds the maximum record size allowed for the output data set. This message may be issued if an output data set has inconsistent RECSIZE and BLKSIZE parameters. For example:

- RECFMOUT=VB and BLKSIZE is less than RECSIZE + 8
- RECFMOUT=V, B, or DB, and BLKSIZE is less than RECSIZE + 4

VSE or CMS: You tried to write a record that is too large for the type of data set you are working with. The record size may be limited by the record format used, the track capacity of a CKD device, the FBA control interval size, or the attributes of a data set defined in the catalog.

User Response: Specify a smaller record size, or change the output data set definition.

**DIT3540I Block size (blksize) not multiple of record
size (reclen)**

Explanation: A block consists of one or more records. The block size must be the same as the record size, or a multiple of the record size.

User Response: Specify a different block size or a different record size.

DIT3550I Record size (nnnnn) invalid for FIXED,nnnnn output

Explanation: The record size encountered is not compatible with the output specifications.

User Response: Correct the input or output specification or use the SET PAD= function to adjust the input records.

DIT3560I Inconsistent record size (nnnnn) for FIXED,nnnnn input

Explanation: The record size encountered is not compatible with the input specifications.

User Response: Correct the input or output specification.

DIT3570I Inconsistent block length field X'xxxx', block length is length

Explanation: While reading variable-length (blocked) input, DITTO found a data block whose block-length field is different from the physical length of the data block.

User Response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

DIT3580I Inconsistent record length field X'xxxx'

Explanation: While reading variable-length (blocked) input, DITTO found a data block whose record-length field contains either zero or a number greater than the physical length of the data block.

User Response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

DIT3590I REXX variable var is empty

Explanation: The indicated REXX variable does not contain a string, or is not defined.

User Response: Initialize the REXX variable in your procedure with the desired value before invoking DITTO.

DIT3600I REXX Stem count var invalid

Explanation: The indicated REXX variable does not contain a valid number, or is not defined.

User Response: Initialize the REXX variable in your procedure with the correct stem count before invoking DITTO.

DIT3610I Incorrect block length field X'xxxx', block nnnn, length nnnn

Explanation: The indicated block does not contain variable format data, or the value in the block descriptor field is different from the physical length of the data block.

User Response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

DIT3620I Incorrect record length field X'xxxx', block nnnn, offset nnnn

Explanation: The indicated block does not contain variable format data, or the value in a record descriptor field is invalid for deblocking.

User Response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

DIT3630I Incorrect spanned record segmentation in block nnnn, record nnnn

Explanation: The indicated block contains a segment of a variable spanned record which cannot be assembled to an entire record.

User Response: Correct the input data, limit processing to stop before the incorrect block, or correct the input specifications.

DIT3700I Key positioning not possible

Explanation: You cannot specify a key position with control interval access or a NONINDEXED VSAM input.

User Response: Remove the key position specification.

DIT3710I Output data set not ESDS

Explanation: You are using the TV function with the *nfiles* parameter, to copy more than one tape data set to a single VSAM data set. The output VSAM data set must be an entry-sequenced data set.

User Response: Specify an ESDS as the output data set.

DIT3720I Inconsistent key length or key position

Explanation: You have copied records from one VSAM key-sequenced data set to another VSAM key-sequenced data set. The output data set has a different key position or key length than the input data set.

User Response: Check that you really intended to change the key position or key length.

DIT3730I VSAM master catalog information not available

Explanation: A VSAM function was invoked, but DITTO does not have information about the master catalog to be used.

User Response: Invoke function VMC before invoking a VSAM function. If running DITTO in command mode, the master catalog information is saved in the session profile. It is used in future DITTO sessions unless you change it with another VMC invocation.

DIT3750I VSAM macro RC rc, Error Code X'xx' yyy

Explanation: A VSAM macro returned with an unexpected code. The message includes the failing macro, the return and error codes, and, for some errors, an additional explanation. For more information, refer to the documentation of your current release of VSAM.

X'xx' is, in hexadecimal:

- The error byte of the ACB if OPEN or CLOSE failed
- The error byte of the RPL if POINT, GET, or PUT failed
- The contents of Register 0 if SHOWCB or TESTCB failed

yyy may present additional information about the error.

User Response: Take corrective action as described in the documentation for the current release of VSAM.

DIT3920I OPEN error X'xx', VSAM/SAM feature not available

Explanation: An attempt was made to open a SAM ESDS without the VSE/VSAM Space Management for SAM feature installed.

User Response: Provide the correct VSAM environment for the function.

DIT3930D OPEN warning code X'xx' yyy. Continue ? Y or N

Explanation: VSAM OPEN returned a warning code indicated in the message. DITTO lets you decide whether to ignore the condition or to exit from the function. For more information, refer to the documentation of your current release of VSAM.

User Response: Take corrective action as described in the documentation for the current release of VSAM. Enter Y to continue, or N to exit from the function.

DIT3950I Recovery from invalid RBA (nnnnn) failed

Explanation: VSAM rejected the RBA you entered. An attempt to locate the first record with a higher RBA also failed.

User Response: Enter an RBA matching the beginning of a record.

DIT3961I SHOWCAT for input/output data set failed, RC rc

Explanation: A VSAM SHOWCAT macro failed (giving return code rc) while searching for information about the specified data set. For an explanation of these codes, refer to the documentation for the current release of VSAM. If the SHOWCAT macro failed for a VSAM alternate index operation, it may show incorrect return codes because of VSAM recovery action.

User Response: Take the appropriate action based on the explanation of the codes.

DIT3962I SHOWCAT for input/output data set failed, RC rc, Error Code rtcd

Explanation: A VSAM SHOWCAT macro failed (giving return code rc) while searching for information about the specified data set. The VSAM return code value is rtcd. For an explanation of these codes, refer to the documentation for the current release of VSAM. If the SHOWCAT macro failed for a VSAM alternate index operation, it may show incorrect return codes because of VSAM recovery action.

User Response: Take the appropriate action based on the explanation of the codes.

DIT3970I DIV req request error, RC nn, Reason Code rsn

Explanation: A data-in-virtual macro has failed. The message includes the type of macro request and the reason code. For an explanation of the return code and reason code, refer to the documentation for the current release of your MVS system.

User Response: Take the appropriate action based on the explanation of the codes.

DIT4010I Input/Output tape on unit is vol1, not volser.

Explanation: The tape that the operator mounted is not the volume requested.

User Response: Ensure that you are using the correct volumes, check for incorrectly specified volume serial numbers, and rerun the function.

DIT4050A Mount next input tape on *unit*, then press ENTER

Explanation: The end of an input tape has been reached, and there is a continuation tape to be mounted. The system waits for a reply.

User Response: Mount a new tape and then press Enter, or enter QUIT, CANCEL, EXIT, E0J to terminate.

DIT4070D Tape volume to be labeled on *unit*. Enter ' *volser*', Cancel - U

Explanation: Standard label output was specified, but the named tape volume is not labeled. The system waits for a reply.

User Response: Enter a serial identification to be used for the volume, or enter U to cancel.

DIT4090D Unexpired file *name date* on tape *unit,vol1*. Ignore - I, Cancel - U

Explanation: The expiration date on the volume mounted as the output tape was not reached. The system waits for a reply.

Operator Response: Ensure that the tape can be used for output and enter I to continue, or enter U to cancel.

DIT4100A Mount next output tape on *unit*, then press ENTER

Explanation: The end of an output tape has been reached, but there is still more output to write. The system waits for a reply.

User Response: Mount a new tape and then press Enter. Note that the tape is rewound but not unloaded, if prevented from unload by the system. Enter QUIT, CANCEL, EXIT, E0J to terminate.

DIT4110A Ready *unit*, then press ENTER

Explanation: The unit *unit* is not ready. The system waits for a reply.

User Response: Ensure that the correct device is attached and assigned. Ready the device and press Enter. Enter QUIT, CANCEL, EXIT, E0J to terminate.

DIT4120A Unprotect tape *unit*, press ENTER when ready

Explanation: You attempted to write on a protected tape. The system waits for a reply.

User Response: Unprotect the tape, or change the tape, and press Enter. Enter QUIT, CANCEL, EXIT, E0J to terminate.

DIT4130D Manual unload/rewind detected on *unit*, press ENTER to rewind and continue

Explanation: A manual operator intervention occurred on the indicated unit. The system waits for a reply.

User Response: Press Enter to rewind and continue, or enter QUIT, CANCEL, EXIT, E0J to terminate.

DIT4140D More input volsers required. Enter ' *volser*', Cancel - U

Explanation: More input volumes are required than specified for the function. The system waits for a reply.

User Response: Enter a serial identification to be used for the next input volume, or enter U to cancel.

DIT4149I SCALE percentage is not supported on device *nnnn*

Explanation: You have specified an Initialize Tape (INT) function with a SCALE parameter greater than zero, but the device does not support scaling capacity. SCALE is valid only for 3590 devices; it is not valid for a 3590 running in 3489E emulation mode.

User Response: Either specify a SCALE parameter of zero, or omit the SCALE parameter.

DIT4150D More output volsers required. Enter ' *volser*', Cancel - U

Explanation: More output volumes are required than specified for the function. The system waits for a reply.

User Response: Enter a serial identification to be used for the next output volume, or enter U to cancel.

DIT5170I Cannot connect to DB2, CAF request RC *xx*, Reason Code *xx*

Explanation: DITTO failed to connect to DB2. The indicated Call Attach Facility request did not complete successfully.

User Response: Contact your DB2® administrator.

DIT5180I Unexpected RC *xx* returned from IEFSSREQ request, token *nnnnnn*

Explanation: A subsystem request to retrieve OAM storage group information from SMS failed with an unexpected return code.

User Response: Contact your system support.

DIT5190I Function terminated by OAM environment error

Explanation: DITTO failed to establish connection with OAM.

User Response: Contact system support.

DIT5200I OAM req failed, RC rc, Reason Code rsn

Explanation: The requested OAM function failed.

User Response: For an explanation of the return code and reason code, refer to the documentation for the current release of MVS.

DIT5210I Directory request failed, DB2 not accessible

Explanation: The requested OAM function failed. DITTO could not access DB2.

User Response: Contact your DB2 administrator.

DIT5220I Directory request failed, DB2 privilege not granted

Explanation: Your user ID is not authorized to access the object directory tables as required.

User Response: Contact your DB2 administrator. For a list of tables, refer to the *DITTO/ESA Installation and Customization Guide*.

DIT5230I Directory request failed, unexpected SQL RC rc

Explanation: The requested OAM function failed. An unexpected return code was received from SQL.

User Response: Contact your DB2 administrator.

DIT5240I STORE failed, collection not found or definition denied

Explanation: An OSREQ STORE request failed because the target collection could not be found, or the user is not authorized to define a new one.

User Response: Check your installation SMS specifications for correct class values if necessary.

DIT5250I Directory request failed with RC rc, trying again...

Explanation: The requested OAM function failed with an SQL return code indicating a locked DB2 resource. This is an information message only. DITTO continues and reissues the SQL request.

DIT5260I Directory request failed with SQL RC rc, DB2 resource locked

Explanation: The requested OAM function failed with an SQL return code indicating a locked DB2 resource.

User Response: Rerun the function. If the problem persists, contact your DB2 administrator.

DIT5290I Skip count exceeds bytes byte object size

Explanation: You asked to skip more bytes than the object contains. You have effectively asked DITTO not to print any data.

User Response: Specify a lower skip count value.

DIT5300I No object header found on input, specify collection and object name

Explanation: The input data do not contain object header records, thus no defaults for collection and object name are available.

User Response: Specify the collection and object name and rerun the function. To create input data with header records, use functions OS or OV with the HEADER option.

DIT5340I STOR/MGMT class value overridden by defaults

Explanation: An OSREQ STORE has completed, but it could not use the Storage Class or Management Class that you specified. Default values for the collection were used instead.

User Response: Check your installation SMS specifications for correct class values if necessary.

DIT5370I SQL RC nn, volume table not found or not in DFP 3.3.1 format

Explanation: Running an OAM function for a specific optical volume requires a volume table in the DFP 3.3.1 (fmid JDP3347) format.

User Response: Upgrade DFP to level 3.3.1 (fmid JDP3347) and run the volume table conversion job CBR SAMMG. Check also the name of your Optical Configuration Database. If not CBROAM (the IBM supplied name), use the DITTO profile parameter OAMDBASE=dbase. Refer to the *DITTO/ESA Installation and Customization Guide*.

DIT5380I Input data not in object dump format

Explanation: An input record was found with unexpected contents, or end of input occurred before the end of an object.

User Response: Correct the input data and rerun the function.

DIT5390I *nnn* object(s) could not be dumped

Explanation: Backup of one or more objects failed. DITTO issues this message at the end of the function. Message DIT5200I shows the OAM error information for each failing object.

User Response: Check the DITTO output for message DIT5200I and take appropriate actions.

DIT5500I Library request *request* failed, RC *rc*, RSCD *rscd*

Explanation: A library request failed, with the indicated return code and reason code.

User Response: For more information, refer to the documentation for the current release of VSE.

DIT5700I Input data invalid for load

Explanation: The data on the input tape is not in DUMP or DDR format, or the input CMS file is not of the specified type.

User Response: Ensure that the input is valid for DITTO load. If the input is a CMS file, specify the correct type (DDR, FULLDDR, or loadfile), then rerun the function.

DIT5720I Decode error encountered, error code *ec*

Explanation: DITTO is unable to expand DDR data from compact format when restoring to DASD. *ec* is the return code from the decoding routine.

User Response: Create the input again. If this is not possible, contact your system support.

DIT5900I Format of DB field incorrect at offset *nnnn*

Explanation: The format of the indicated double-byte character set field is incorrect. DITTO found a double-byte field that has an odd length.

DITTO issues this message for only the first incorrect field it finds. It prints this message at the end of the record, and replaces the last character with an EBCDIC blank (X'40'). The function continues.

DIT5910I Mixed field ends before end of DB subfield at offset *nnnn*

Explanation: The double-byte character set (DBCS) subfield, at the offset shown, was not reset at the end of a mixed DBCS and EBCDIC field. For example, there may be no shift in (X'0F') character marking the end of a double-byte field in mixed DBCS and EBCDIC data.

DITTO issues this message for only the first incorrect field it finds. It prints this message at the end of the record. The function continues.

User Response: Correct the field definition in the FMT command. Rerun the function.

DIT5920I One or more records with incorrect DB fields found

Explanation: DITTO has detected incorrect double-byte character set fields in one or more records.

DITTO prints this message at the end of the listing. It indicates that DITTO issued one or more error messages. The function continues.

User Response: Check the listing for message DIT5900I or DIT5910I to find the incorrect record.

DIT6001I Minimum requirement to run DITTO is MVS/ESA

Explanation: To run DITTO/ESA under MVS, you need MVS/ESA.

User Response: Use a DITTO level capable of running on your current system level.

DIT6002I Minimum requirement to run DITTO is VM/ESA

Explanation: To run DITTO/ESA under CMS, you need VM/ESA.

User Response: Use a DITTO level capable of running on your current system level.

DIT6003I Minimum requirement to run DITTO is VSE/ESA

Explanation: To run DITTO/ESA under VSE, you need VSE/ESA Version 2 Release 1 or later.

User Response: Use a DITTO level capable of running on your current system level.

DIT6010I Minimum requirement to run DITTO is DFP 3.3

Explanation: To run DITTO/ESA under MVS, you need MVS/DFP Version 3 Release 3 or later.

DIT6040I Load of module '*name*' failed, module not found

Explanation: DITTO cannot find a module which is required for your DITTO invocation. This module may have been deleted or protected (for example, if the functions supported by this module are not needed at your installation).

User Response: If you need to use this function, contact your system support.

DIT6050I Successfully recovered from ABEND

Explanation: An ABEND occurred while DITTO was running. Depending on the circumstances, a dump may be produced.

User Response: Check for any system error or DITTO message and take corrective actions. If the problem persists, contact your system support.

DIT6060I DITTO encountered an internal error - CODE *nnnn*

Explanation: DITTO encountered an internal problem. The error is identified by the DITTO error code.

User Response: Contact your system support.

DIT6070I DITTO abended with CODE *code* - REASON *reason*

Explanation: An abend occurred while DITTO was running. *code* is the MVS abend code, VSE cancel code, or CMS abend code; *reason* contains additional information that is available for some cancel codes and abend codes.

User Response: Refer to the documentation for the current release of MVS, VSE or CMS.

DIT6081I Load of module *xxxxxxx* failed, ABEND code *nnnn-*nnn**

Explanation: An attempt to load the specified module failed with an unexpected error. *nnnn-*nnn** is the MVS ABEND and REASON code.

User Response: Refer to the accompanying system message or the system code, and take appropriate action.

DIT6082I NUCXLOAD of module *xxxxxxx* failed, RC *rrrr*

Explanation: An attempt to load the specified module failed with an unexpected error. *rrrr* is the NUCXLOAD return code.

User Response: Take appropriate action as described in the documentation for your current level of CMS.

DIT6083I CDLOAD of module *xxxxxxx* failed, RC *rrrr*

Explanation: An attempt to load the specified module failed with an unexpected error. *rrrr* is the CDLOAD return code.

User Response: Take appropriate action as described in the documentation for your current release of VSE.

DIT6090I Access to REXX variable pool failed, *xxx* R15 *nn*

Explanation: During an attempt to establish connection with the REXX variable pool system services failed.

User Response: Contact your system support.

DIT6100I Unexpected code returned from REXX variable interface, RC *nn*

Explanation: During an attempt to read from or write to the REXX variable pool the REXX system interface returned an unexpected result.

User Response: Contact your system support.

DIT6110I Permanent I/O error - device not attached

Explanation: An attempt was made to run a function on a device that is not attached or on a disk that is not linked.

User Response: Refer to the system console log for detailed information about the devices that are attached and linked. Attach and link the device if it is free; otherwise, use another device.

DIT6131I I/O error on *unit*, ECB *ecb*, CSW *csw*, sense 0-3 *sense*

Explanation: An unrecoverable I/O error occurred on the indicated device.

User Response: Ensure that you are using the correct device, and that it is set online correctly. Scan the console log for an I/O error message, and save it. If the error persists, give the message text to your system support.

DIT6132I I/O error on *unit*, DIAG*xx*, CC *cc*, RC *rc*, CSW *csw*, sense 0-3 *sense*

Explanation: The CP DIAGNOSE instruction, which is used to perform I/O operations, failed because of an unrecoverable I/O error. The reason could be a hardware error, a device not linked or attached correctly, or an attempt to access a device not supported by the function used.

User Response: Ensure that you are using the correct device and that it is linked and attached correctly. Then rerun the function. For information about error codes, refer to the documentation for the device and your current level of CP. If the problem persists, contact your system support.

DIT6133I I/O error on *unit*, CCBCOM *com*, CSW *csw*, sense 0-3 *sense*

Explanation: An unrecoverable I/O error occurred on the indicated device.

User Response: Ensure that you are using the correct device, and that it is set online correctly. Scan the console log for an I/O error message, and save it. If the error persists, give the message text to your system support.

DIT6140I Spool file read error, DIAGNOSE X'14' CC *cc*, RC *rc*

Explanation: A spool processing error occurred in the spool file currently being browsed.

User Response: Take appropriate action for DIAGNOSE X'14' as described in the documentation for the current level of CP. If the problem persists, contact your system support.

DIT6150I Virtual Console error, DIAGNOSE X'08' RC *rc*

Explanation: An error occurred during the execution of a diagnose X'08' instruction (virtual console function).

User Response: The return code is the numeric value expressed in the message describing the error. For example, if error message DMKCPM045E is issued, CP sets a return code of 45. For more information, refer to the documentation for the current level of CP.

DIT6160I CMS *macro* problem, R15 *rc*

Explanation: A CMS macro failed. The macro and the return code are given in the message. For more information, refer to the documentation for the current level of CMS.

User Response: Take corrective action according to the return code. Rerun the function with new parameters.

DIT6170I Error processing file *filename filetype*, *macro* RC *rc*

Explanation: An error occurred while opening, reading, or writing a CMS file. A possible reason is that recent changes were applied to the disk holding the indicated file; thus, it must be reaccessed. A write error is most probably caused by a disk full condition. The return code indicated is returned by an FSOPEN, FSREAD, or FSWRITE macro. For more information, refer to the documentation for the current level of CMS.

User Response: Check which disk contains the file, and reaccess that disk. Provide sufficient CMS disk space for output files. Then rerun the function.

DIT6180I Cannot access catalog disk, no free file mode found

Explanation: No free file mode is available to access a CMS/DOS disk,

User Response: Release at least one disk not used in the DITTO session, then rerun the function.

DIT6190I CMS/DOS environment not available, *command* RC *rc*

Explanation: An attempt to use a function that needs an active CMS/DOS environment failed because of one of following:

- The SET DOS ON command, which is issued internally by DITTO, was unsuccessful.
- CMS/DOS is not available at your installation.

User Response: Make sure it is possible to set DOS on for your user ID, and that CMS/DOS is available in your installation.

DIT6200I CMS/DOS disk access failed, *command* RC *rc*

Explanation: DITTO failed to establish the CMS/DOS environment. The message indicates the failing command and the return code received.

User Response: Make sure you have linked and accessed the correct disks. In VSAM functions, check the master catalog setup with the DITTO function VMC.

Refer also to the *VM/ESA CMS Command Reference* for the command return code, and take appropriate actions.

DIT6210I CMS formatted volume *volser* invalid for *func* function

Explanation: SAM and VSAM functions and functions that access a VTOC require OS (MVS) or DOS (VSE) OS formatted disks, but the specified volume is in CMS format.

User Response: Make sure you have linked and accessed the correct disk, then try the function again.

DIT6230I Function *xxxx* not possible on read/only disk

Explanation: A DITTO function was called; however, the disk is linked in read/only mode.

User Response: Check whether the disk can be linked in read/write mode. If so, relink the disk in read/write mode, then rerun the function.

DIT6500I Dynamic action failed, RC *rc*, ERR *ec*, INFO *ic ddname*

Explanation: The indicated SVC99 (allocation) operation failed.

User Response: For explanations of *rc*, *ec*, and *ic*, refer to the documentation for the current release of MVS.

DIT6510I Locate macro failed, RC *rc*, REG0 X'xxxxxxxx'

Explanation: Unexpected return code from the LOCATE macro.

User Response: For an explanation of the return code and register contents, refer to the documentation for the current release of MVS.

DIT6520I TIOT scan error, *ddname* not found

Explanation: An error occurred while scanning the MVS control block TIOT. This is probably a DITTO error.

User Response: Exit DITTO and rerun the function. If the problem persists, contact your system support.

DIT6530I DSCB/JFCB read for *xxxxxxx* failed, RC *rc*

Explanation: Unexpected return code when reading the DSCB or JFCB.

User Response: For an explanation of the return code, refer to the descriptions of the macros OBTAIN and RDJFCB in the documentation for the current release of MVS.

DIT6540I UCB scan error, RC *rc*, Reason Code *rsn*

Explanation: A UCB scan operation (to determine the device) failed with an unexpected return code.

User Response: For an explanation of the return code, refer to the description of the UCBSKAN macro in the documentation of your current release of MVS.

DIT6550I TSO Service Facility call error - RC *rc*, Reason Code *rsn*

Explanation: An unexpected return code was received from the TSO Service Facility. The TSO command was not executed.

User Response: For an explanation of the return code and reason code, refer to *TSE/E Programming Services*.

DIT6750I action VTOC failed, RC *rc*

Explanation: A VTOC access operation failed with an unexpected error. The failing action can be OPEN or READ. The return code indicates the type of the problem.

User Response: VSE: If the return code is 012 or 004, issue the ROD command. Run the LVTOC Utility Program and check for errors. If the problem persists, report it to your system support.

CMS: Invoke the LISTDS command for the volume and check for errors. If the problem persists, contact your system support.

DIT6760I Extent setup failed, EXTENT macro RC *rc*

Explanation: DITTO encountered an EXTENT macro problem. *rc* is the EXTENT macro return code.

User Response: Contact your system support.

DIT6770I Label access failure - action RC *rc*

Explanation: Accessing the label area failed with an unexpected error. The message shows the failing action and the return code.

User Response: Issue the ROD command. Run the LSERV Utility Program and check for errors. If the problem persists, contact system support.

DIT6780I Request canceled, see console messages for error information

Explanation: An input/output request failed. Additional VSE system error messages identify the cause of the error.

User Response: Refer to the console log for associated error messages and codes. For a description of the error codes, refer to the documentation for the current release of VSE.

DIT6790I Spool access error, XPCB RC *rc*, FDBK *rtcd*

Explanation: DITTO encountered a spool access problem. *rc* and *rtcd* are the XPCB return and feedback codes.

User Response: Contact your system support.

DIT6800I Label area not available, update in progress

Explanation: Accessing the system label area failed. The label area is locked by another task.

User Response: Check the system console for a pending reply, then rerun the function. If the problem persists, contact your system support.

DIT6810I Assigning *unit failed*, *macro RC nn*

Explanation: A device could not be assigned; the indicated macro failed with an unexpected error.

User Response: Check the status of the device. Take appropriate action as described in the documentation for the indicated macro. If the problem persists, contact your system support.

DIT6820I *macro for unit failed, R15 rc*

Explanation: The indicated macro for the indicated unit failed with an unexpected return code.

User Response: Take appropriate action as described in the documentation for the indicated macro. If the problem persists, contact your system support.

DIT6830I *macro failed, R15 rc*

Explanation: The indicated macro failed with an unexpected return code.

User Response: Take appropriate action as described in the documentation for the indicated macro. If the problem persists, contact your system support.

DIT6840I *macro failed, RC rc RSCD rscd*

Explanation: The indicated macro failed with an unexpected return code and reason code.

User Response: Take appropriate action as described in the documentation for the indicated macro. If the problem persists, contact your system support.

DIT9990I **DITTO problem - message *nnnn* not found in table**

Explanation: DITTO cannot find the message number *nnnn* in the message table. This is probably a DITTO error.

User Response: Contact your system support.

Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023
U.S.A.

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

Notices

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

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

COPYRIGHT LICENSE:

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This User's Guide and Reference documents intended Programming Interfaces that allow the customer to write programs to obtain the services of DITTO/ESA.

Trademarks

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

| | |
|----------------------------|---------|
| Advanced Function Printing | MVS |
| AFP | MVS/DFP |
| BookManager | MVS/ESA |
| CICS | OS/390 |
| DB2 | RACF |
| DFSMS | VM/ESA |
| DFSMS/MVS | VSE/ESA |

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

Bibliography

DITTO/ESA Publications

DITTO/ESA Installation and Customization Guide, GC26-9598

Introducing DITTO/ESA, GH19-8222

DITTO/ESA Licensed Program Specifications, GH19-8223

DITTO/ESA Reference Summary, SX11-6110

DITTO/ESA User's Guide and Reference, SH19-8221

Program Directory for DITTO/ESA for MVS, GI10-0425

Program Directory for DITTO/ESA for VM, GI10-0438

Program Directory for DITTO/ESA for VM NLV Japanese, GI10-0427

Related Publications for MVS

OS/390 ISPF Planning and Customizing, SC28-1298

OS/390 MVS Initialization and Tuning Guide, SC28-1751

OS/390 MVS Initialization and Tuning Reference, SC28-1752

OS/390 MVS JCL Reference, GC28-1757

DFSMS/MVS® Access Method Services for Integrated Catalog Facility, SC26-4906

DFSMS/MVS OAM Application Programmer's Reference, SC26-4917

DFSMS/MVS Using Data Sets, SC26-4922

DFSMS/MVS Using Magnetic Tapes, SC26-4923

OS/390 Security Server (RACF) Command Language Reference, SC28-1919

OS/390 Security Server (RACF) Security Administrator's Guide, SC28-1915

SMP/E Messages and Codes, SC28-1738

SMP/E Reference, SC28-1806

SMP/E User's Guide, SC28-1740

TSE/E Command Reference, SC28-1881

TSE/E Programming Services, SC28-1875

Related Publications for VM

VM/ESA CMS Command Reference, SC24-5776

VM/ESA CMS File Pool Planning, Administration, and Operation, SC24-5751

VM/ESA Planning and Administration, SC24-5750

VM/ESA SFS and CRR Planning, Administration, and Operation, SC24-5649

ISPF and ISPF/PDF Planning and Customizing for VM, SC34-4305

VM/ESA Connectivity Planning, Administration, and Operation, SC24-5448

VM/ESA Version 2.1 and 2.2

VM/ESA Service Guide, SC24-5749

VM/ESA System Messages and Codes, SC24-5784

VM/ESA VMSES/E Introduction and Reference, SC24-5747

VM/ESA Version 2.3

VM/ESA Service Guide, GC24-5838

VM/ESA System Messages and Codes, GC24-5841

VM/ESA VMSES/E Introduction and Reference, GC24-5837

Related Publications for VSE

VSE/ESA Administration, SC33-6605

VSE/ESA System Macro User's Guide, SC33-6615

VSE/ESA Guide to System Functions, SC33-6611

VSE/ESA Installation, SC33-6604

VSE/ESA Planning, SC33-6603

VSE/ESA System Control Statements, SC33-6613

VSE/ESA Messages and Codes, SC33-6607

VSE/VSAM Commands, SC33-6631

VSE/VSAM User's Guide and Application Programming, SC33-6632

General Publications

Field Engineering Programming System General Information, G229-2228

Softcopy Publications for DITTO/ESA

The following collection kits contain DITTO/ESA or related product publications in BookManager® format:

MVS Collection, SK2T-0710

OS/390 Collection, SK2T-6700

VM Collection, SK2T-2067

VSE Collection, SK2T-0060

To view the latest DITTO/ESA product publications in Acrobat PDF format, visit the DITTO/ESA site on the World Wide Web at:

<http://www.software.ibm.com/ad/ditto/>

Glossary

This publication includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the *American National Dictionary for Information Processing*, copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.

The following terms are defined as they are used in this manual. If you do not find the term you are looking for, refer to the Index or to the *IBM Dictionary of Computing*, SC20-1699.

ANSI definitions are preceded by an asterisk.

A

across format. A DITTO dump format that has the same appearance as a system dump. To get this format for your listings, use the DUMP=ACROSS parameter of the SET function.

action bar. The area at the top of a window that contains choices that give a user access to choices available in that window.

ASCII. * American National Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8-bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

B

block. * A collection of contiguous records recorded as a unit. Blocks are separated by interblock gaps and each block may contain one or more logical records.

A block is a *physical record* but it may be composed of several contiguous *logical records* or parts of *logical records*.

blocking. The process of combining two or more *logical records* in one block.

BLP. Bypass label processing.

buffer. A portion of storage used to hold input or output data temporarily.

C

cards. (1) Actual punched cards that can be read with a card reader. (2) Lines in a batch job. (3) Card image data. In VSE, card image data is read from SYSIPT or the terminal and written to SYSPCH. In CMS, card image data is read from the virtual reader address 00C and written to the virtual punch address 00D. In MVS, card image data is read from SYSIN and written to SYSPUNCH.

cartridge. A storage device that consists of magnetic tape, on supply and takeup reels, in a protective housing.

catalog. (1) * A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, and blocking factors. (2) * To enter information about a file or library into a catalog.

character fill. * To insert as often as necessary into a storage medium the representation of a specified character that does not itself convey data but may delete unwanted data.

CKD. Count key data, a disk architecture.

CMS/VSE mode. A VSE/DITTO server that is controlled from a CMS client DITTO session.

collection. See *object collection*.

command mode. A way to enter a single DITTO command, with all necessary parameters, from outside a DITTO session. You can invoke DITTO in command mode from a selection list or within a REXX procedure. After the command is executed, you leave DITTO.

Common User Access (CUA) architecture. Guidelines for the dialog between a person and a workstation or terminal.

console. See *operator console*.

control interval. A fixed-length area of direct access storage in which VSAM stores records and creates distributed free space. Also, in a key-sequenced data set or file, the set of records pointed to by an entry in the sequence-set index record. The control interval is the unit of information that VSAM transmits to or from direct access storage. A control interval always comprises an integral number of *physical records*.

Glossary

cylinder. The tracks of a disk storage device that can be accessed without repositioning the access mechanism.

D

data definition (DD) statement. A job control statement describing a data set associated with a specific job step.

data definition name (ddname). The name of a data definition (DD) statement that corresponds to a data control block that contains the same name.

data-in-virtual data set. A data set that is considered a continuous stream of uninterrupted data.

DBCS. Double-byte character set.

deblocking. The process of making each *logical record* of a block available for processing. Contrast with *blocking*.

E

EBCDIC. * Extended Binary-Coded Decimal Interchange Code. A coded character set consisting of 8-bit coded characters.

end of file (EOF). An internal label or special record immediately following the last record of a file, signaling the end of that file. It may include control totals for comparison with counts accumulated during processing.

end of volume (EOV). An internal label, or 2 consecutive tape marks, that follows and indicates the end of a set of data contained in a volume.

extended help. Help text that applies to an entire function or panel. Contrast with *field help*.

extent. Continuous space on a disk or diskette that is occupied by or reserved for a particular data set, data space, or file.

F

FBA. Fixed-block architecture, a disk architecture supported by VSE and CMS but not by MVS.

field help. Help text that applies to a specific entry field, parameter, or term. Contrast with *extended help*.

file, tape. See *tape file*.

fill. See *character fill*.

full-screen mode. A type of DITTO session, in which you can use full-screen panels to select functions,

specify parameters, browse data, and update data. Some DITTO functions are available only in full-screen mode.

H

header label. An internal label, immediately preceding the first record of a file, that identifies the file and contains data used in file control.

I

interactive (processing). Pertaining to an application in which each entry calls for a response from a system or program, as in an inquiry system or an airline reservation system. An interactive system may also be conversational, implying a continuous dialog between the user and the system.

interpret. To print on a punched card the characters corresponding to the hole patterns punched in the card.

K

KSDS. Key-sequenced data set.

L

label. A record that identifies a volume on tape, disk, or diskette or that identifies a file on the volume.

label, header. See *header label*.

line mode. A type of DITTO session, in which you enter commands and are prompted for parameters. After the command is executed, you remain in DITTO.

linear data set. A named linear string of data, stored in such a way that it can be retrieved or updated in 4096-byte units.

logical record. (1) A record from the standpoint of its content, function, and use rather than its physical attributes, that is, a record defined in terms of the information it contains. (2) In VSAM, a unit of information normally pertaining to a single subject; a logical record is the user record requested of or given to the data management function.

O

OAM. Object Access Method.

object. In OAM, data consisting of a contiguous stream of bytes of arbitrary length, up to a maximum of 15 megabytes. The contents and internal structure of an object are defined by the application using it.

Objects are increasingly used in image processing and other advanced application areas.

object collection. A collection of user-defined objects.

operator console. A functional unit containing devices that are used for communications between a computer operator and a computer.

P

parameter string. Two or more parameters separated by commas or blanks.

physical record. A record whose characteristics depend on the manner or form in which it is stored, retrieved, or moved. A physical record may contain all or part of one or more *logical records*.

Q

QSAM. Queued Sequential Access Method, an access method for sequential data sets.

R

record size. The number of characters or bytes in a record.

RRDS. Relative record data set.

S

SAF. System Authorization Facility, a system facility for security.

shift-out/shift-in. Characters that mark the beginning and end of double-byte character set data. The shift-out character is X'0E', and the shift-in character is X'0F'.

staging file. A file created by the LOAD function, which you can use for subsequent LOAD operations.

SMS. Storage Management Subsystem.

standard label. A fixed-format record that identifies a tape that is part of a multivolume tape file.

T

tape file. One or more data blocks between tape marks.

tape mark. An indicator record that marks the beginning or the end of a set of data such as a file.

tape unit. A device containing a magnetic tape drive, magnetic heads, and associated controls.

tape volume. A reel of magnetic tape.

U

unit, tape. See *tape unit*.

updown format. A DITTO dump format with hexadecimal representation of data in 3 lines: character, zone, and number. To get this format for your listings, use the DUMP=UPDOWN parameter of the SET function.

utility program. A computer program in general support of computer processes; for example, a diagnostic program, a trace program, a sort program.

V

volume. (1) Tape volume: A reel of magnetic tape. (2) Disk volume: A disk pack or part of a disk storage module.

volume serial number (volser). A number in a volume label assigned when a volume is prepared for use in the system.

volume table of contents (VTOC). A table on a direct access volume that describes each file on the volume.

VSAM. Virtual Storage Access Method.

Glossary

Index

Special Characters

/ for searching 34, 37, 41
 * symbol in CMS file ID 120
 * symbol in library member name 205, 206
 * symbol in names 188, 225, 231, 232, 239, 257, 314, 322
 ** symbol in names 188, 225, 231, 232, 239, 257, 314, 322
 % symbol in CMS file ID 120
 % symbol in names 188, 205, 206, 225, 231, 232, 239, 257, 314, 322
 = to repeat a command 57

Numerics

3200 printer 246
 5550 printer 246

A

absolute disk address 88, 246
 across format 4, 33, 37, 41
 action bar 15
 Actions function key 15
 ADDKEY command in DTE function 43
 addresses, disk
 absolute and relative 88, 246
 AL tape processing 62, 65
 allocation
 tape 65
 altering
 See updating
 ANSI tape labels, creating 248
 APB function
 how to use 32
 ASCII
 converting to and from EBCDIC 242
 not supported for TFA, TFD, TLT, TTC, TU 81, 243
 parameter of SET 242
 ATL (automated tape library) 66
 automated tape library 66

B

back up
 OAM objects 138
 batch mode
 control statements 25
 description 11
 errors 25
 example
 disk 99

batch mode (*continued*)
 example (*continued*)
 library 131
 OAM 141
 sequential data sets 107
 tape 84
 VSAM 114
 XXX function 25
 syntax 24
 batch partition (VSE) 12
 Bkwd function key 34, 37, 41
 BLKFACTOR keyword, obsolete 351, 352
 BLP tape processing 63, 65
 BQ function
 See BS function
 browse functions 31
 BS function
 batch example 107, 155
 format 152
 how to use 106
 BSF function
 format 156
 how to use 68
 BSQ function
 See BS function
 BSR function
 format 158
 how to use 68
 BT function
 batch example 161
 command mode example 161
 format 159
 how to use 83
 BTP function
 See BT function
 buffer functions 35, 38
 BV function
 batch example 164
 format 162
 how to use 114
 BVS function
 See BV function
 bypass label processing 63, 65

C

cancel conditions
 batch mode 25
 full-screen mode 19
 line mode 21
 Cancel function key 16

Index

- CAPS command in edit functions 46
- cards
 - copying 145
 - functions 143—145
 - input and output 144
 - interpreting 145
 - interrupting 144
- CASE command in edit functions 46
- catalog
 - functions 51—60
 - IDCAMS parameters 58
 - printing a list of entries 52
- CC function
 - format 165
 - how to use 145
- CCL function
 - format 165
 - how to use 144
- CD function
 - See CP function
- CHANGE command in edit functions 43
 - syntax determined by SET CMDSYNTAX 243
- changes to DITTO/ESA in each release xv
- character format 4, 33, 37, 41
- CI function
 - See CC function
- CIACCESS 111
- CICS transaction (VSE) 12
- CISIZE keyword
 - obsolete for sequential functions 352
- CKD devices 88, 95
 - transfer files to FBA 120
- CL function
 - batch example 166
 - format 166
 - how to use 145
- client server mode 14
- CLIPBOARD function 48
- CLIPBOARD parameter of MB 35, 38
- closing a tape 236
- CMDLINE function 17
- CMDSYNTAX, parameter of SET 243
- CMS files
 - browsing 32
 - copying records 119
 - dumping 119
 - functions 117—121
 - input and output 118
 - loading 119
 - overview of functions 117
 - printing records 118
- CMS/VSE mode 14
- command mode
 - description 11
 - example 23
 - disk 98
 - library 131
- command mode (*continued*)
 - example (*continued*)
 - OAM 140
 - tape 84
 - VSAM 114
 - how to use 21
- commands
 - See functions
- Common User Access interface 15
- comparing tapes 82
- compatibility with previous releases xix
- console
 - displaying a disk record 94
 - displaying a tape record 77
- control interval access 111
- conversion mode, tape 67
- converting
 - See also translating
 - ASCII and EBCDIC 242
 - not supported for TFA, TFD, TLT, TTC, TU 81, 243
 - browse formats 33, 37
 - decimal and hexadecimal numbers 17
 - edit formats 41
- COPY function 74, 93, 112, 129, 137
- copying
 - cards 145
 - CMS files 119
 - library members 130
 - OAM objects 138
 - sequential records 106
 - tape data across a network 81
 - tape records 80
 - VSAM records 114
- count-key data (CKD) devices 88, 95
 - transfer files to FBA 120
- CP function
 - format 167
 - how to use 144
- CQ function
 - See CS function
- creating
 - sequential (SAM, QSAM) records 106
 - tape records 83
 - VSAM records 114
- CRetrieve function key 16
- CS function
 - batch example 170
 - format 168
 - how to use 145
- CSQ function
 - See CS function
- CSR, scroll value 33, 37, 41
- CT function
 - format 171
 - how to use 145

- CUA interface 15
 - CV function
 - batch example 174
 - format 173
 - how to use 145
 - CVS function
 - See CV function
 - CYLHD, parameter of SET 88, 246
 - cylinder head values 88, 246
- D**
- data create functions
 - sequential data sets 106
 - tape 83
 - VSAM 114
 - data header, omitting 246
 - data security erase 83, 191
 - data-check errors, correcting 71
 - data-in-virtual data sets
 - browsing 32
 - DATA, scroll value 33, 37, 41
 - DATAHDR, parameter of SET 246
 - DB function
 - how to use 32
 - DBCS printing
 - DBCSPRT parameter 246
 - FMT function 192
 - output formats 4
 - DBCSPRT, parameter of SET 246
 - DCN function
 - format 176
 - how to use 94
 - DD function
 - See DP function
 - DDD function
 - See DP function
 - DDR tapes, loading 120
 - decimal
 - converting numbers to hexadecimal 17
 - defining catalog entries 54
 - deleting
 - library member 131
 - delimiter, EOD
 - See EOD, parameter of SET
 - density, tape 67
 - DID function
 - command mode example 177
 - format 177
 - how to use 97
 - Directory function key 128
 - disk
 - absolute and relative addresses 88, 246
 - browse 32
 - end-of-file record, writing 96
 - extents 88, 92, 95, 128
 - disk (*continued*)
 - fullpack actions 89
 - functions 87—99
 - getting data from a buffer 93
 - input 88, 95
 - label area, VSE 92
 - overview of functions 87
 - printing a list of files 89
 - printing records 95
 - putting data into a buffer 35
 - scanning 97
 - update 93
 - volume ID 97
 - VTOC information 89
 - display formats
 - examples 4
 - when browsing 33, 37
 - when editing 41
 - DITSETUP
 - required for VSE command mode 21, 23
 - DIV data sets
 - browsing 32
 - DLA function
 - format 178
 - how to use 92
 - double tape mark 69
 - double-byte character set data 4, 192, 246
 - DOWN command 34, 37, 41
 - DP function
 - batch example 99, 180
 - command mode example 98
 - format 179
 - how to use 95
 - DPD function
 - See DP function
 - DRL function
 - format 181
 - how to use 93
 - DRS function
 - batch example 184
 - format 182
 - how to use 97
 - DSE function
 - batch example 185
 - format 185
 - how to use 92
 - DTE function
 - how to use 40
 - DU function
 - how to use 93
 - dump formats 4, 246
 - DUMP function
 - command mode example 187
 - format 187
 - how to use 119

Index

DUMP, parameter of SET 246

DVB function
how to use 32

DVT function
batch example 189
format 188
how to use 89

DX function 17

dynamic partition under VSE 24

E

EBCDIC

converting to and from ASCII 242
not supported for TFA, TFD, TLT, TTC, TU 81,
243

EDIT command in LE function 47

edit functions 39

editing data at your terminal
See edit functions

end-of-data delimiter

See EOD, parameter of SET

end-of-file record, disk 96

end-of-volume mark, tape 69, 70

enhancements to DITTO/ESA in each release xv

environment, operating system 3

EOD, parameter of SET

default 246

for cards 246

for tape 246

used to process tapes beyond EOVS 70

EOF function

format 190

how to use 96

EOJ function

batch example 25

format 190

line mode example 21

EOT 70

EOV 69, 70

erasing

OAM object 140

print output 36

spool files 36

tape records 83

errors

batch mode 25

data check, correcting 71

full-screen mode 19

line mode 21

messages 353—372

ERT function

format 191

how to use 83

EXEC statement 24

Exit function key 16

expiration date
tape 65

extended help 16

Extents function key 128

extents, disk 88, 92, 95, 128

F

FB function

how to use 32

FBA devices 88, 96

transfer files to CKD 120

FD function

See FP function

FDD function

See FP function

field help 15

files

See CMS files

FIND command 34, 37, 41

fixed-block-architecture (FBA) devices 88, 96

transfer files to CKD 120

FL function

how to use 119

FMT function

batch example 193

format 192

format

examples 4

tape 68

when browsing 33, 37

when editing 41

FP function

command mode example 194

format 194

how to use 118

FPR function

See FP function

FREE command, TSO 66

FS function

how to use 119

FSF function

format 195

how to use 68

FSN function

format 196

how to use 18

FSR function

format 197

how to use 68

FSY function

format 198

how to use 18

FT function

command mode example 199

FT function (*continued*)
 copying tapes across a network 81
 format 198
 how to use 119

FTP function
 See FT function

full-screen mode
 browse functions 31
 description 11
 edit functions 39
 errors 19
 example 19
 news function 16
 online help 15
 panel format 15
 starting 11
 switching to and from line mode 18
 tutorial 16
 unique functions 17

fullpack, disk 89

function keys
 Actions 15
 Bkwd 34, 37, 41
 Cancel 16
 CRetrieve 16
 description 16
 Directory 128
 Exit 16
 Extents 128
 Fwd 34, 37, 41
 Help 15, 16
 keys help 16
 Left 35, 42
 menu key 16
 Next 38
 Previous 38
 Process 56, 91, 126, 129, 136
 Purge 19, 36
 Rfind 34, 38, 42
 Rgtright 55, 90, 125, 135
 Right 35, 42
 Split (ISPF) 18
 Statistics 60
 Sublibs 128
 Swap 18
 Zoom 35, 42, 74, 93, 112

functions
 browse 31—38
 card 143—145
 catalog 51—60
 CMS 117—121
 disk 87—99
 edit 39—50
 library 123—131
 OAM 133—141
 obsolete 349

functions (*continued*)
 overview 3
 sequential (SAM, QSAM) 101—108
 tape 61—85
 VSAM 109—115

FV function
 format 200
 how to use 119

FVS function
 See FV function

Fwd function key 34, 37, 41

G

GDG 59
 generic names 188, 205, 206, 225, 231, 232, 239,
 257, 314, 322
 GET function 35, 38, 48, 74, 93, 112, 129, 137
 GETFILE command in LE function 47
 glossary 377—379
 GO, in line mode 20

H

HALF, scroll value 33, 37, 41
 HEADERPG, parameter of SET 246
 headers, omitting 246
 help for help 15
 Help function key 15, 16
 help index 16
 hexadecimal
 across format 4, 33, 37, 41
 converting numbers to decimal 17
 updown format 4, 33, 37, 41

I

IDCAMS parameters 58

II function
 format 201
 how to use 145

index, online help 16

initializing tapes 82

INOUT used for tape open 63, 65

input
 cards 144
 disk 88, 95
 library 124
 sequential data sets 102
 tape 62
 data-check errors 71
 VSAM 110

INPUT used for tape open 63, 65

INT function
 batch example 82, 203
 command mode example 203

Index

INT function (*continued*)

format 201
how to use 82

interpreting cards

II function 201

interrupting a DITTO function

line mode 21

ISPF

split and swap functions 18

J

Japanese

See *also* DBCS printing

Katakana characters 17, 247

job parameters (VSE) 12

JOIN command in LE, VE function 47

jump facility 17

K

Kanji characters

See DBCS printing

Katakana characters 17, 247

keys

See *also* function keys

help 16

VSAM KSDS 111

KEYS function 17

keyword syntax

command mode 21

keywords, obsolete 349

KSDS (key-sequenced data set) 111

L

label area, VSE 92

label value for tapes 65

LANGUAGE, parameter of SET 247

layout of tapes 68

LB function

how to use 32

LC function

batch example 204

format 204

how to use 130

LD function

See LP function

LDEL function

batch example 131, 205

command mode example 131

format 205

how to use 131

LDL function

batch example 207, 217

format 206

LDL function (*continued*)

how to use 124

LE function

See *also* LDEL function

format 207

how to use 40

Left function key 35, 42

level of DITTO

LVL function 221

LF function

how to use 130

LI function

See LC function

library

browsing members 32

copying members 130

deleting members 131

editing members 40

functions 123—131

getting data from a buffer 129

input 124

member list 124

output 124

printing a list of members 124

printing members 130

putting data into a buffer 35, 38, 48, 49

renaming members 131

status list 127

updating members 129

LIBSERV command (VSE) 64

limiting tape capacity 202

line commands

DVT 91

LDL 126

LSL 129

ODL 136

SCS and VDL 57

line length for print output 247

line mode

description 11

errors 21

example 21

starting 19

switching to and from full-screen mode 18

linear data sets

control interval access 112

lines per page, print output 247

LL function

batch example 209, 214

format 208

how to use 130

LOAD function

command mode example 211, 212

format 210

how to use 119

load point, positioning tape at 68
 LOCATE command 34, 37, 41
 syntax determined by SET CMDSYNTAX 243
 LP function
 batch example 131, 213
 command mode example 131
 format 212
 how to use 130
 LR function
 format 214
 how to use 131
 LS function
 format 215
 how to use 130
 LSL function
 format 217
 how to use 127
 LT function
 batch example 219
 format 218
 how to use 130
 LU function
 how to use 129
 LV function
 batch example 221
 format 220
 how to use 130
 LVL function
 format 221
 LX function
 command mode example 223
 format 222
 how to use 130

M

MAP command in DTE function 43
 map, tape 71
 master catalog, VSAM 111
 MAX, scroll value 33, 37, 41
 MB function
 how to use 32
 memory
 browsing 32
 putting data into a buffer 35
 Menu function key 16
 messages
 list 353—372
 selecting language for 247
 mm (tape mode) values 67
 MODE keyword
 obsolete, except for SP (MVS) 352
 mode, tape 67
 moving
 VSE Librarian members 208

N

NBLKS keyword
 obsolete for TP, TRS 352
 network, copying a tape across 81
 news function 16
 Next function key 38
 nfiles parameter
 specifying EOD 246
 NL tape processing 65
 notices 373

O

OAM
 backing up objects 138
 browsing objects 32
 copying objects 138
 erasing objects 140
 functions 133—141
 getting data from a buffer 137
 object list 134
 printing a list of objects 134
 printing objects 137
 putting data into a buffer 38
 restoring objects 139
 updating objects 136
 OB function
 how to use 32
 objects
 See OAM
 obsolete functions and keywords 349
 OD function
 See OP function
 ODL function
 batch example 225
 command mode example 140
 format 224
 how to use 134
 OE function
 batch example 141, 226
 command mode example 140
 format 226
 how to use 140
 online help 15
 OO function
 command mode example 228
 format 227
 how to use 138
 OP function
 batch example 141, 229
 format 229
 how to use 137
 open for write, tape 63, 65
 operating system levels 3

Index

- options, SET function 242
- OQ function
 - See OS function
- OS function
 - batch example 231
 - format 230
 - how to use 138
- OTL function
 - See OS function
- OU function
 - format 231
 - how to use 136
- OUTIN used for tape open 63, 65
- OUTMODE keyword, obsolete 352
- output
 - cards 144
 - library 124
 - print
 - See print output
 - sequential data sets 102
 - tape 62
 - VSAM 110
- OV function
 - command mode example 233
 - format 232
 - how to use 138
- overview
 - all DITTO functions 3
 - browse functions 31
 - card functions 143
 - catalog functions 51
 - CMS functions 117
 - disk functions 87
 - edit functions 39
 - library functions 123
 - OAM functions 133
 - sequential (SAM, QSAM) functions 101
 - tape functions 61
 - VSAM functions 109
- P**
- PAD, parameter of SET 247
- page length, print output 247
- PAGE, scroll value 33, 37, 41
- PAGESIZE, parameter of SET 247
- PAGESKIP, parameter of SET 247
- PANELID function 17
- panels
 - selecting language for 247
- partition, dynamic 24
- partitioned data set
 - See sequential data sets
- PB function
 - example 19
 - how to use 26, 32
- PBN (physical block number) 96
- permanent tape assignment 68, 248
- PFSHOW function 17
- physical block number (PBN) 96
- physical records
 - disk, processing 87
- positioning a tape 68
- PREFIX command in edit functions 46
- Previous function key 38
- previous releases of DITTO xix
- previous releases of DITTO/ESA, changes to xv
- PRINT function 17, 36
- print output
 - browsing 32
 - destination 26, 247
 - format 4
 - line length 247
 - page size 247
 - purging 19, 36
 - putting data into a buffer 35
 - translation 247
- printing
 - browsed data 36, 38
 - cards 144
 - CMS files 118
 - DBCS characters 4, 192, 246
 - destination 26, 247
 - disk records 95
 - library member 130
 - OAM object 137
 - sequential data sets 105
 - SYSLST disk files 105
 - SYSLST tape files 79
 - tape files 79
 - translation tables 247
 - VSAM data sets 113
- PRINTLEN, parameter of SET 247
- PRINTOUT, parameter of SET 26, 247
- Process function key 56, 91, 126, 129, 136
- processing options 242
- processing tapes beyond double tape mark 70
- PRTRTRANS, parameter of SET 247
- pull-down menu 15
- PURGE function 36
- Purge function key 19, 36
- purging
 - See erasing
- PUT function 35, 38, 48, 74, 93, 112, 129, 137
- PVT function
 - batch example 234
 - command mode example 234
 - format 234
 - how to use 91

Q

QC function
 See SC function

QD function
 See SP function

QI function
 See SC function

QO function
 See SO function

QP function
 See SP function

QQ function
 See SS function

QSAM
 See sequential data sets

QSAMIN 102

QSAMIN ddname 102

QSAMOUT 102

QSAMOUT ddname 102

QT function
 See ST function

question-and-answer format 19

QV function
 See SV function

R

RD function 36

reader file
 browsing 32

RECLIMIT, parameter of SET 248

RECOVER command in edit functions 47

RECSIZE keyword
 obsolete for FT function 352

relative disk address 88, 246

relative record data set 111

release, changes to DITTO/ESA in each xv

renaming
 library member 131

replacing
 VSE Librarian members (when copying or moving) 209

RESET command in edit functions 47

RESET, parameter of SET 242

restoring
 OAM objects 139

return codes
 batch mode 24
 command mode 22
 DUMP function 187
 LOAD function 210

REW function
 batch example 84
 command mode example 84
 format 235

REW function (*continued*)
 how to use 68

rewind a tape 235
 and unload 236

REXX
 command mode 21

REXX stem variable
 from library member 222
 from tape records 310
 from VSAM records 337
 print output 26, 247
 to library member 341
 to tape file 342
 to VSAM data set 344

Rfind function key 34, 38, 42

Rgtright function key 55, 90, 125, 135

Right function key 35, 42

routing output 26, 247

RP function 36

RRDS (relative record data set) 111

RUN function
 format 236
 how to use 68

running DITTO
 batch mode 24
 command mode 21
 full-screen mode 11
 line mode 19

S

SAM
 See sequential data sets

SAVE command in edit functions 49

saving data in a buffer 35, 38, 48, 49

SC function
 batch example 238
 command mode example 238
 format 237
 how to use 106

scaling maximum tape capacity 202

scanning
 disk records 97
 tape records 78

scratch tapes 63, 65

SCS function
 batch example 241
 command mode example 23
 format 239
 how to use 52

SDP function
 See SP function

SDSKIN dlbl 102

SDSKOUT dlbl 102

sending tapes across a network 81

Index

- sequence number, tape 65
- sequential data sets
 - batch example 107
 - copying records 106
 - creating records 106
 - functions 101—108
 - input and output 102
 - printing 105
 - VSAM-managed SAM files 102
- SET function
 - batch example 99
 - command mode example 23, 98, 140
 - format 242
 - line mode example 21
 - overview 10
 - returning to default values 242
- SF function
 - command mode example 250
 - format 249
 - how to use 106
- SFA function
 - format 251
 - how to use 105
- SFD function
 - batch example 253
 - format 252
 - how to use 105
- SI function
 - See SC function
- SL function
 - format 254
 - how to use 106
- SL tape processing 62, 63, 65
- slot value 111
- SMS
 - See OAM
- SO function
 - batch example 257
 - format 256
 - how to use 139
- SP function
 - batch example 259
 - format 258
 - how to use 105
- SPB function
 - how to use 32
- SPL function
 - introduction 260
- SPLIT command in LE, VE function 47
- Split function key (ISPF) 18
- SPLTJOIN command in LE, VE function 47
- spool file
 - browsing 32
- SPR function
 - See SP function
- SPS function
 - introduction 260
- SS function
 - batch example 107, 264
 - format 261
 - how to use 106
- SSQ function
 - See SS function
- ST function
 - batch example 268
 - copying tapes across a network 81
 - format 265
 - how to use 106
- staging file 120
- stand-alone VSE support 27
- START keyword, obsolete 352
- starting DITTO
 - command mode 21
 - full-screen mode 11
 - line mode 19
- Statistics function key 60
- status information for a library 127
- STP function
 - See ST function
- sublibraries of a library 127
- Sublibs function key 128
- SUBMIT command in LE function 47
- SV function
 - batch example 108, 271
 - format 269
 - how to use 106
- SVS function
 - See SV function
- Swap function key 18
- synonyms for functions and keywords 349
- syntax diagrams
 - batch mode 151
 - command mode (keyword syntax) 151
 - command mode (parameter syntax) 150
 - line mode 150
- SYSIN
 - input for card functions (MVS) 144
- SYSIPT
 - assignment 27
 - input for card functions (VSE) 144
- SYSLST
 - assignment 27
 - disk files 105
 - tape files 79
- SYSOUT class
 - browsing print output 233
 - print destination 26, 247
- SYSPCH
 - assignment 27
 - output for card functions (VSE) 144

SYSPRINT
 print destination 26, 247
 required in batch mode under MVS 24
 SYSPUNCH
 output for card functions (MVS) 144
 system files (VSE) 27

T

tailoring DITTO 242

tape

- allocating input and output 65
- ANSI labels, creating 248
- assignment 68, 248
- block size 68
- browse 32
- bypass label processing 63, 65
- closing 236
- comparison function 82
- conversion mode 67
- copying across a network 81
- copying records 80
- creating records 83
- data-check errors 71
- density 67
- determining contents 71
- dumping files to 119
- file, defined 68
- functions 61—85
- getting data from a buffer 74
- initializing 82
- input 62
- large blocks 68
- layout 68
- loading files from 119
- map 71
- mark
 - converted during copy 81
 - double 69, 70
 - separates tape files 68
 - writing 69
- mode 67
- output 62
- permanent assignment 68, 248
- positioning 68
- printing records 79
- putting data into a buffer 35
- rewinding 235
- rewinding and unloading 236
- scaling maximum capacity 202
- scanning 78
- scratch 63, 65
- summary, how to get 71
- temporary assignment 68, 248
- update 74

- TAPEASGN, parameter of SET 68, 248
- TAPEIN tlibl 64
- TAPELBL, parameter of SET 248
- TAPEOUT tlibl 64
- Task Selection menu 11
- TB function
 - how to use 32
- TC function
 - format 272
 - how to use 80
- TCN function
 - format 274
 - how to use 77
- TD function
 - See TP function
- TDD function
 - See TP function
- TDV function
 - See TP function
- temporary tape assignment 68, 248
- TF function
 - command mode example 276
 - copying tapes across a network 81
 - format 275
 - how to use 80
- TFA function
 - batch example 277
 - format 277
 - how to use 79
- TFD function
 - format 278
 - how to use 79
- TFT function
 - batch example 280
 - command mode example 280
 - format 279
 - how to use 80
- TI function
 - See TC function
- TL function
 - batch example 282
 - format 281
 - how to use 80
- TLB function
 - batch example 284
 - format 283
 - how to use 71
- TLO function
 - See SO function
- TLT function
 - batch example 286
 - format 285
 - how to use 80
- TMP function
 - batch example 288
 - command mode example 288

Index

TMP function (*continued*)
 format 287
 how to use 71

TP function
 batch example 84
 command mode example 84
 format 289
 how to use 79

TPD function
 See TP function

TPV function
 See TP function

TQ function
 See TS function

translating
 messages and panels 247
 unprintable characters 247

translation tables
 SET function 247

transmitting tapes across a network 81

TRL function
 format 291
 how to use 75

TRS function
 batch example 295
 format 293
 how to use 78

truncated records 247

TS function
 batch example 299
 copying tapes across a network 81
 format 296
 how to use 80

TSOPRINT keyword, obsolete 352

TSQ function
 See TS function

TT function
 batch example 84, 301
 command mode example 84
 format 300
 how to use 80

TTC function
 batch example 303
 format 302
 how to use 82

TTR function
 batch example 306
 format 304
 how to use 80

TU function
 how to use 74
 special considerations 74

tutorial 16

TV function
 command mode example 309
 copying tapes across a network 81

TV function (*continued*)
 format 307
 how to use 80

TVS function
 See TV function

TX function
 command mode example 311
 format 310
 how to use 80

TYPE keyword
 obsolete for TMP function 351

U

unit, tape 65

unload tape after rewinding 236

unprintable characters, translation of 247

UP command 34, 37, 41

updating
 disk records 93
 library members 129
 OAM objects 136
 tape blocks 74, 75
 VSAM records 112, 113

updown format 4, 33, 37, 41

user storage
 browsing 32

V

VB function
 how to use 32

VC function
 batch example 313
 format 312
 how to use 114

VD function
 See VP function

VDL function
 batch example 315
 format 314
 how to use 52

VDP function
 See VP function

VDSKIN dlbl 110

VDSKOUT dlbl 110

VE function
 format 316
 how to use 40

VER function
 format 316

version of DITTO
 VER function 316

VF function
 format 317
 how to use 114

- VI function
 - See VC function
 - viewing data at your terminal
 - See browse functions
 - VL function
 - batch example 131, 319
 - command mode example 131
 - format 318
 - how to use 114
 - VMC function
 - format 320
 - how to use 111
 - VO function
 - batch example 141
 - format 321
 - how to use 139
 - volser, tape 65
 - VP function
 - batch example 114, 324
 - command mode example 114
 - format 323
 - how to use 113
 - VPR function
 - See VP function
 - VQ function
 - See VS function
 - VRL function
 - See VRU function
 - VRU function
 - format 325
 - how to use 113
 - VS function
 - batch example 329
 - format 326
 - how to use 114
 - VSAM
 - browsing data sets 32
 - browsing data-in-virtual data sets 32
 - control interval access 111
 - copying records 114
 - editing data sets 40
 - functions 109—115
 - getting data from a buffer 112
 - input 110
 - master catalog 111
 - output 110
 - overview of functions 109
 - printing a list of data sets 52
 - printing records 113
 - processing order 111
 - putting data into a buffer 35, 38, 48, 49
 - updating data sets 112
 - VSAM-managed SAM files 102
 - VSAMIN ddname 110
 - VSAMOUT ddname 110
 - VSE stand-alone support 27
 - VSE system files 27
 - VSE/DITTO from CMS 14
 - VSQ function
 - See VS function
 - VT function
 - batch example 333
 - copying tapes across a network 81
 - format 330
 - how to use 114
 - VTOC, disk 89
 - VTP function
 - See VT function
 - VU function
 - how to use 112
 - VV function
 - batch example 336
 - format 334
 - how to use 114
 - VVS function
 - See VV function
 - VX function
 - command mode example 338
 - format 337
 - how to use 114
- ## W
- writing
 - disk EOF record 96
 - tape marks 69, 339
 - WTM function
 - format 339
 - how to use 69
- ## X
- XD function 17
 - XL function
 - command mode example 341
 - format 341
 - how to use 130
 - XT function
 - format 342
 - how to use 80
 - XV function
 - format 344
 - how to use 114
 - XXX function
 - batch example 25
 - format 345
 - line mode example 21

Index

Z

Zoom function 35, 42, 74, 93, 112

We'd Like to Hear from You

DITTO/ESA
User's Guide and Reference
Release 3
Publication No. SH19-8221-02

Please use one of the following ways to send us your comments about this book:

- Mail—Use the Readers' Comments form on the next page. If you are sending the form from a country other than the United States, give it to your local IBM branch office or IBM representative for mailing.
- Fax—Use the Readers' Comments form on the next page and fax it to this U.S. number: 800-426-7773.
- Electronic mail—Use one of the following network IDs:
 - IBMLink: HLASMPUB at STLVM27
 - Internet: COMMENTS@VNET.IBM.COM

Be sure to include the following with your comments:

- Title and publication number of this book
- Your name, address, and telephone number if you would like a reply

Your comments should pertain only to the information in this book and the way the information is presented. To request additional publications, or to comment on other IBM information or the function of IBM products, please give your comments to your IBM representative or to your IBM authorized remarketer.

IBM may use or distribute your comments without obligation.

Readers' Comments

DITTO/ESA

User's Guide and Reference

Release 3

Publication No. SH19-8221-02

How satisfied are you with the information in this book?

| | Very Satisfied | Satisfied | Neutral | Dissatisfied | Very Dissatisfied |
|--------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Technically accurate | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to find | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to understand | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Well organized | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Applicable to your tasks | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Grammatically correct and consistent | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Graphically well designed | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Overall satisfaction | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Please tell us how we can improve this book:

May we contact you to discuss your comments? Yes No

Name

Address

Company or Organization

Phone No.



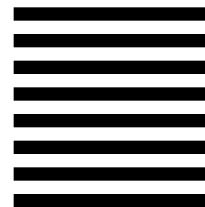
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department HHX/H1
555 Bailey Avenue
SAN JOSE, CA 95141-1099



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5655-103 (OS/390 or MVS)
5648-099 (VSE)
5654-029 (VM)



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

SH19-8221-02





DTTO/ESA

User's Guide and Reference

Release 3