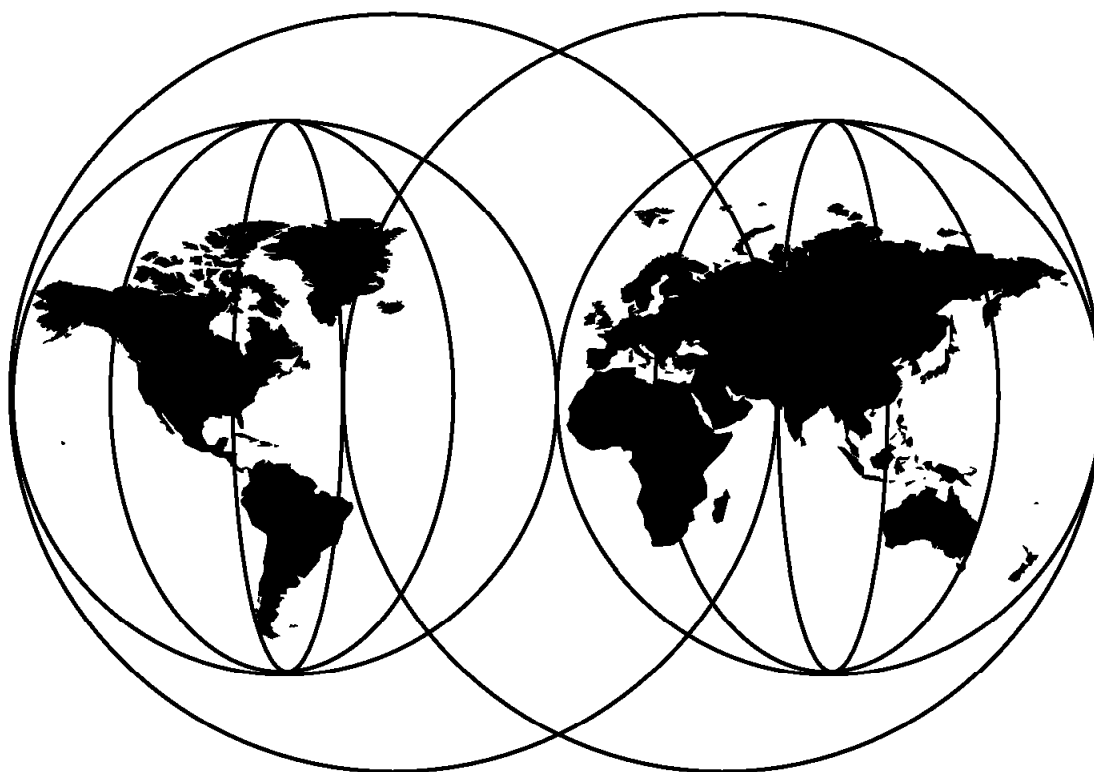




ABCs of OS/390 System Programming

Volume 1

*P. Rogers, G. Capobianco, D. Carey, N. Davies, L. Fadel, K. Hewitt,
J. Oliveira, F. Pita, A. Salla, V. Sokal, Y. F. Tay, H. Timm*



International Technical Support Organization

www.redbooks.ibm.com



International Technical Support Organization

SG24-5597-00

ABCs of OS/390 System Programming
Volume 1

April 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, "Special Notices" on page 303.

First Edition (April 2000)

This edition applies to OS/390 Version 2 Release 8, Program Number 5647-A01, and to all subsequent releases and modifications.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, New York 12601-5400

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

© **Copyright International Business Machines Corporation 2000. All rights reserved.**

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

Contents

| | |
|--|------|
| Figures | vii |
| Tables | xiii |
| Preface | xv |
| The team that wrote this redbook | xv |
| Comments welcome | xvi |
| Chapter 1. Introduction to OS/390 | 1 |
| 1.1 Introduction to OS/390 fundamentals | 2 |
| 1.1.1 OS/390 evolution | 3 |
| 1.1.2 OS/390 software and hardware | 4 |
| 1.1.3 OS/390 elements and features | 6 |
| 1.2 OS/390 products requiring customization | 8 |
| 1.2.1 Resource Access Control Facility (RACF) | 9 |
| 1.2.2 Data Facility Storage Management Subsystem (DFSMS) | 10 |
| 1.2.3 ADSTAR Distributed Storage Management (ADSM) | 11 |
| 1.2.4 Transmission Control Protocol/Internet Protocol (TCP/IP) | 12 |
| 1.2.5 System Modification Program Extended (SMP/E) | 13 |
| 1.2.6 Resource Management Facility (RMF) | 14 |
| 1.3 OS/390 components requiring customization | 15 |
| 1.3.1 System Management Facility (SMF) | 16 |
| 1.3.2 Virtual Lookaside Facility (VLF) | 18 |
| 1.3.3 Time Sharing Option/Extended (TSO/E) | 20 |
| 1.3.4 Workload Manager (WLM) | 21 |
| 1.3.5 UNIX System Services | 22 |
| 1.3.6 OS/390 contents | 24 |
| 1.4 Role of a system programmer | 26 |
| 1.5 OS/390 system programmer functions | 28 |
| 1.5.1 System programmer and OS/390 operations | 31 |
| 1.5.2 Requirements for install | 34 |
| 1.6 Installing OS/390 | 36 |
| 1.6.1 Maintenance or service using SMP/E | 36 |
| 1.6.2 Choosing an install package | 37 |
| 1.6.3 Installing OS/390 using ServerPac | 38 |
| 1.6.4 Installing OS/390 using CBPDO | 40 |
| 1.6.5 Installing OS/390 using fee-based packages | 42 |
| Chapter 2. OS/390 Storage Concepts | 43 |
| 2.1 OS/390 Address Spaces | 44 |
| 2.1.1 Subsystem definitions | 45 |
| 2.2 Processor Storage Overview | 47 |
| 2.3 Storage managers | 49 |
| 2.3.1 Virtual Storage Manager | 50 |
| 2.3.2 Virtual Storage | 51 |
| 2.4 Address space storage | 54 |
| 2.4.2 Real Storage Manager | 64 |
| 2.4.3 Auxiliary Storage Managers | 66 |
| 2.4.4 Paging and Swapping | 68 |
| 2.5 Program execution | 71 |

| | |
|---|-----|
| Chapter 3. S/390 hardware and I/O management | 73 |
| 3.1 Data center | 74 |
| 3.2 Logical processor structure | 76 |
| 3.3 Channel subsystem | 77 |
| 3.3.1 Start Subchannel (SSCH) logic | 78 |
| 3.3.2 SAP logic | 78 |
| 3.3.3 Interrupt processing | 78 |
| 3.4 Multiple paths to a device | 79 |
| 3.5 Device number | 80 |
| 3.6 Subchannel number | 81 |
| 3.6.1 Subchannel number II | 82 |
| 3.7 Unit address | 83 |
| 3.8 Control unit function | 85 |
| 3.9 Parallel channel | 86 |
| 3.10 ESCON channel | 87 |
| 3.11 S/390 ESCON concepts | 89 |
| 3.11.1 ESCON link | 89 |
| 3.11.2 Link address | 90 |
| 3.12 Mapping device number in unit address | 91 |
| 3.13 ESCON CUADD function | 92 |
| 3.14 ESCON director (switch) | 94 |
| 3.15 ESCON director matrix | 96 |
| 3.16 ESCON channel-to-channel (CTC) | 97 |
| 3.17 Fiber Connection (FICON) | 99 |
| 3.18 How to display channel types | 101 |
| 3.19 S/390 connectivity | 102 |
| 3.20 System-related I/O work flow | 103 |
| 3.21 What is HCD | 105 |
| 3.21.1 HCD functions | 107 |
| 3.21.2 Dynamic I/O reconfiguration | 109 |
| 3.21.3 Dynamic reconfiguration concepts | 111 |
| 3.22 IODF data set | 113 |
| 3.22.1 Catalog considerations | 114 |
| 3.22.2 Production IODF backup | 114 |
| 3.22.3 IODF placement | 114 |
| 3.22.4 Multiple configurations in a single IODF | 115 |
| 3.22.5 Definition order | 116 |
| 3.23 HCD primary option menu | 117 |
| 3.23.1 How to create a new work IODF | 119 |
| 3.23.2 Defining configuration data | 120 |
| 3.24 Operating system definition | 121 |
| 3.24.1 How to define an operating system | 122 |
| 3.24.2 EDT and esoterics | 123 |
| 3.24.3 Select new configuration for EDTs | 125 |
| 3.24.4 How to define an EDT | 126 |
| 3.24.5 How to define an EDT | 127 |
| 3.24.6 Define EDT identifier | 128 |
| 3.24.7 How to add an esoteric | 129 |
| 3.24.8 Add an esoteric | 130 |
| 3.25 Defining switches | 131 |
| 3.25.1 How to define a switch | 132 |
| 3.26 Defining processors | 133 |
| 3.26.1 Basic mode | 133 |
| 3.26.2 LPAR mode | 133 |
| 3.26.3 Defining processors with HCD | 134 |

| | | |
|---|--|------------|
| 3.26.4 | Information required to define a processor | 135 |
| 3.26.5 | How to define a processor | 136 |
| 3.26.6 | Work with partitions | 137 |
| 3.26.7 | How to add a partition to a CEC | 138 |
| 3.26.8 | Channel types | 139 |
| 3.26.9 | Reconfigurable channels | 141 |
| 3.26.10 | Shared or EMIF channels | 142 |
| 3.26.11 | Information required to add channels | 143 |
| 3.26.12 | Work with channel paths | 144 |
| 3.27 | How to add a channel path | 145 |
| 3.28 | Channel path access list | 146 |
| 3.29 | Channel path candidate list | 146 |
| 3.30 | How to add a control unit | 147 |
| 3.31 | Information required to define a control unit | 148 |
| 3.31.1 | Adding a control unit | 149 |
| 3.32 | Defining a 3990-6 control unit | 150 |
| 3.32.2 | Select processor for control unit | 152 |
| 3.32.3 | Defining a 3990-6 control unit | 153 |
| 3.32.4 | Defining processor attachment data | 154 |
| 3.33 | Information required to define a device | 155 |
| 3.33.1 | S/390 device numbering | 157 |
| 3.33.2 | How to define an I/O device | 159 |
| 3.33.3 | Device / Processor Definition | 161 |
| 3.33.4 | Defining CSS definitions for a device | 162 |
| 3.33.5 | Define device to operating system | 163 |
| 3.33.6 | Define operating system device parameters | 164 |
| 3.33.7 | Assign device to an esoteric | 166 |
| 3.34 | Defining a NIP console | 167 |
| 3.35 | Build a production IODF | 168 |
| 3.35.1 | Production IODF created | 170 |
| 3.36 | Activating a configuration with HCD | 171 |
| 3.37 | How to display active IODF from the HCD panels | 172 |
| 3.38 | How to display active IODF from an MVS console | 173 |
| 3.39 | How to display a device status | 174 |
| 3.40 | Producing configuration reports | 176 |
| 3.40.2 | HCD graphical reports | 177 |
| 3.41 | Hardware Configuration Manager (HCM) | 179 |
| Chapter 4. TSO/E, ISPF, and JCL | | 181 |
| 4.1 | Installation and customization facilities | 182 |
| 4.1.1 | Time Sharing Option/Extended (TSO/E) | 184 |
| 4.1.2 | Interactive System Productivity Facility (ISPF) | 193 |
| 4.1.3 | JCL streams and jobs | 211 |
| 4.1.4 | DD statement parameters | 217 |
| 4.1.5 | Submitting a job | 223 |
| 4.1.6 | Introduction to the Spool Display and Search Facility (SDSF) | 229 |
| Chapter 5. MVS delivery and installation | | 235 |
| 5.1 | Introduction to OS/390 | 236 |
| 5.1.1 | Advantages of OS/390 | 236 |
| 5.1.2 | OS/390 delivery options | 237 |
| 5.1.3 | ServerPac service level | 237 |
| 5.1.4 | CBPDO service level | 238 |
| 5.1.5 | System and installation requirements | 239 |
| 5.1.6 | Reviewing your current system | 240 |

| | |
|---|------------|
| 5.1.7 The driving system and the target system | 241 |
| 5.2 Installing OS/390 using ServerPac | 242 |
| 5.2.1 Load RIM tape | 243 |
| 5.2.2 Installing the CustomPac dialogs | 244 |
| 5.2.3 Receiving the ServerPac order | 245 |
| 5.2.4 Order Receive panel | 246 |
| 5.2.5 Generate Jobstream panel | 247 |
| 5.2.6 Select order to install | 248 |
| 5.2.7 Installation dialog | 249 |
| 5.2.8 Selecting a configuration for the order | 250 |
| 5.2.9 Define the installation variables | 251 |
| 5.2.10 Defining SMP/E ZONE names | 252 |
| 5.2.11 Define system layout | 253 |
| 5.2.12 Dslist line command | 254 |
| 5.2.13 Select command | 255 |
| 5.2.14 Adding user-defined data sets | 256 |
| 5.2.15 Display physical volumes | 257 |
| 5.2.16 Define alias-to-catalog relationships | 258 |
| 5.2.17 Define system-specific alias names | 259 |
| 5.2.18 Run the ServerPac-provided installation jobs | 260 |
| 5.2.19 Save Configuration panel | 262 |
| | |
| Appendix A. Fundamentals of OS/390 | 263 |
| A.1 List of Base Elements | 263 |
| A.2 List of Optional Elements | 271 |
| | |
| Appendix B. System Programmer Customization Examples | 277 |
| B.1 SYS1.PARMLIB Member IEFSSNxx | 277 |
| | |
| Appendix C. System programmers toolbox | 281 |
| | |
| Appendix D. Special Notices | 303 |
| | |
| Appendix E. Related Publications | 305 |
| E.1 IBM Redbooks | 305 |
| E.2 IBM Redbooks collections | 306 |
| E.3 Other resources | 306 |
| | |
| How to get IBM Redbooks | 309 |
| IBM Redbooks fax order form | 310 |
| | |
| Glossary | 311 |
| | |
| IBM Redbooks evaluation | 325 |

Figures

| | | |
|-----|--|----|
| 1. | OS/390: S/390 Server Operating System | 3 |
| 2. | Hardware/Software Requirements | 4 |
| 3. | OS/390 Elements and Features | 6 |
| 4. | Products Requiring System Programmer Customization | 8 |
| 5. | RACF - Resource Access Control Facility | 9 |
| 6. | Data Facility Storage Management Subsystem | 10 |
| 7. | ADSTAR Distributed Storage Management | 11 |
| 8. | Transmission Control Protocol/Internet Protocol | 12 |
| 9. | System Modification Program Extended (SMP/E) | 13 |
| 10. | Resource Management Facility (RMF) | 14 |
| 11. | Components Requiring System Programmer Customization | 15 |
| 12. | System Management Facility (SMF) | 16 |
| 13. | Virtual Lookaside Facility (VLF) | 18 |
| 14. | Time Sharing Option/Extended (TSO/E) | 20 |
| 15. | Workload Manager (WLM) | 21 |
| 16. | UNIX System Services | 22 |
| 17. | OS/390 Contents | 24 |
| 18. | OS/390 Operating System | 26 |
| 19. | OS/390 System Programmer Management Overview | 28 |
| 20. | System Programmer and Operations | 31 |
| 21. | Requirements for Install | 34 |
| 22. | Installing OS/390 | 36 |
| 23. | Installing OS/390 - ServerPac | 38 |
| 24. | Installing OS/390 - CBPDO | 40 |
| 25. | Installing OS/390 Using Fee-Based Packages | 42 |
| 26. | OS/390 address spaces | 44 |
| 27. | Subsystem definitions | 45 |
| 28. | S/390 storage | 47 |
| 29. | Storage managers | 49 |
| 30. | Virtual storage manager | 50 |
| 31. | Virtual storage | 51 |
| 32. | Multiple address spaces | 53 |
| 33. | Address space storage | 54 |
| 34. | Common versus private area | 56 |
| 35. | System Queue Area (SQA/Extended SQA) | 58 |
| 36. | Common service area | 59 |
| 37. | Nucleus | 60 |
| 38. | Region | 61 |
| 39. | Data space | 62 |
| 40. | Multiprogramming and multiprocessing | 63 |
| 41. | Real storage manager | 64 |
| 42. | Auxiliary storage manager | 66 |
| 43. | Frame, slots and pages | 67 |
| 44. | Paging and swapping | 68 |
| 45. | Paging operation | 69 |
| 46. | Swapping operation | 70 |
| 47. | Program execution | 71 |
| 48. | Data center | 74 |
| 49. | Logical processor structure | 76 |
| 50. | Channel subsystem logic | 77 |
| 51. | Multiple paths to a device | 79 |

| | | |
|------|---|-----|
| 52. | Device number | 80 |
| 53. | Subchannel number | 81 |
| 54. | Subchannel number II | 82 |
| 55. | Unit address | 83 |
| 56. | Control unit function | 85 |
| 57. | Parallel channel | 86 |
| 58. | ESCON channel | 87 |
| 59. | S/390 ESCON concepts | 89 |
| 60. | Mapping device number in unit address | 91 |
| 61. | ESCON CUADD function | 92 |
| 62. | ESCON director (switch) | 94 |
| 63. | ESCON director matrix | 96 |
| 64. | ESCON CTC | 97 |
| 65. | FICON | 99 |
| 66. | How to display channel types | 101 |
| 67. | S/390 connectivity | 102 |
| 68. | System-related I/O work flow | 103 |
| 69. | What is HCD | 105 |
| 70. | HCD functions | 107 |
| 71. | Dynamic I/O reconfiguration | 109 |
| 72. | Dynamic I/O device types | 111 |
| 73. | IODF file | 113 |
| 74. | Definition order | 116 |
| 75. | HCD primary option menu | 117 |
| 76. | How to create a new work IODF | 119 |
| 77. | Defining configuration data | 120 |
| 78. | Operating system definition | 121 |
| 79. | How to define an operating system | 122 |
| 80. | EDT and esoteric | 123 |
| 81. | Select new configuration for EDTs | 125 |
| 82. | How to define an EDT | 126 |
| 83. | How to define an EDT | 127 |
| 84. | Define EDT identifier | 128 |
| 85. | How to add an esoteric | 129 |
| 86. | Add an esoteric | 130 |
| 87. | Defining switches | 131 |
| 88. | How to define a switch | 132 |
| 89. | Defining processors | 133 |
| 90. | Information required to define a processor | 135 |
| 91. | How to define a processor | 136 |
| 92. | Work with partitions | 137 |
| 93. | How to add a partition to a CEC | 138 |
| 94. | Channel types | 139 |
| 95. | Reconfigurable channels | 141 |
| 96. | Shared or EMIF channels | 142 |
| 97. | Information required to add channels | 143 |
| 98. | Work with channel paths | 144 |
| 99. | How to add a channel path | 145 |
| 100. | Access and candidate lists | 146 |
| 101. | How to add a control unit | 147 |
| 102. | Information required to define a control unit | 148 |
| 103. | Adding a control unit | 149 |
| 104. | Defining a 3990-6 control unit | 150 |
| 105. | Select processor for control unit | 152 |
| 106. | Defining a 3990-6 control unit | 153 |

| | | |
|------|--|-----|
| 107. | Defining processor attachment data | 154 |
| 108. | Information required to define a device | 155 |
| 109. | S/390 device numbering | 157 |
| 110. | Defining a device | 159 |
| 111. | Device / Processor Definition | 161 |
| 112. | Defining CSS definitions for a device | 162 |
| 113. | Define device to operating system | 163 |
| 114. | Define operating system device parameters | 164 |
| 115. | Assign device to an esoteric | 166 |
| 116. | Defining a NIP console | 167 |
| 117. | Build a production IODF | 168 |
| 118. | Production IODF created | 170 |
| 119. | Activating a configuration with HCD | 171 |
| 120. | How to display active IODF from the HCD | 172 |
| 121. | How to display active IODF from an MVS console | 173 |
| 122. | How to display a device status | 174 |
| 123. | Producing configuration reports | 176 |
| 124. | Hardware Configuration Manager (HCM) | 179 |
| 125. | OS/390 facilities for system programmers | 182 |
| 126. | TSO/E required customization | 184 |
| 127. | TSO/E TCAS start procedure | 186 |
| 128. | TSO/E logon procedure | 187 |
| 129. | TSO/E logon process in a VTAM environment | 189 |
| 130. | TSO/E full-screen logon panel | 190 |
| 131. | TSO/E languages | 192 |
| 132. | ISPF components | 193 |
| 133. | Sample CLIST to start ISPF environment | 195 |
| 134. | ISPF Primary Option menu | 197 |
| 135. | Allocating data sets with ISPF | 198 |
| 136. | ISPF Option 3.2 panel | 200 |
| 137. | Allocate New Data Set panel | 202 |
| 138. | Edit Entry panel - option 2 | 204 |
| 139. | ISPF Edit panel | 205 |
| 140. | ISPF edit - some line commands | 206 |
| 141. | ISPF Edit panel - inserting lines | 207 |
| 142. | ISPF edit - repeating and deleting lines | 208 |
| 143. | ISPF edit - save and cancel | 209 |
| 144. | JCL streams and jobs | 211 |
| 145. | Creating jobs - an introduction to JCL | 213 |
| 146. | The job statement explained | 214 |
| 147. | The EXEC statement | 214 |
| 148. | EXEC statement parameters | 215 |
| 149. | JCL DD statements: DDname, DSN | 217 |
| 150. | The DD statement and DISP parameter | 217 |
| 151. | JCL DD statement parameters: Disp, Unit | 219 |
| 152. | DD statement parameters: Space, Lrecl, blksize | 221 |
| 153. | Submitting a job | 223 |
| 154. | ISPF option 3.4 | 224 |
| 155. | Data set list for MYUSER | 225 |
| 156. | Displayed data set list commands | 225 |
| 157. | Data set member list | 226 |
| 158. | Action option list | 227 |
| 159. | Creating a PDS member using option 3.4 | 227 |
| 160. | SDSF Primary Option menu | 229 |
| 161. | SDSF - Viewing the JES2 output files | 230 |

| | | |
|------|--|-----|
| 162. | SDSF displaying the output job | 231 |
| 163. | JES2 system messages | 232 |
| 164. | SDSF - display active users (DA) | 233 |
| 165. | SDSF output display | 234 |
| 166. | Installation overview | 236 |
| 167. | System and installation requirements | 239 |
| 168. | Reviewing your current system | 240 |
| 169. | The driving and target system | 241 |
| 170. | OS/390 installation | 242 |
| 171. | The CustomPac dialogs | 244 |
| 172. | Receiving the ServerPac order | 245 |
| 173. | Order Receiving panel | 246 |
| 174. | Generate Jobstream panel | 247 |
| 175. | Select order to install | 248 |
| 176. | Installation Dialog panel | 249 |
| 177. | Create Configuration panel | 250 |
| 178. | Installation Variables panel | 251 |
| 179. | Define ZONE Information panel | 252 |
| 180. | Modify System Layout panel | 253 |
| 181. | Data Set List by Product panel | 254 |
| 182. | Logical Volume by Product panel | 255 |
| 183. | List All User Defined Data Sets panel | 256 |
| 184. | Summary of Physical Volumes panel | 257 |
| 185. | Define Catalog Data Set Name panel | 258 |
| 186. | SSA to Catalog panel | 259 |
| 187. | Installation Jobs panel | 260 |
| 188. | Generate File Tailored Jobs panel | 261 |
| 189. | Save Configuration panel | 262 |
| 190. | Sample 1 - DFSMSdss compress data sets on VOLSER=xxxxx | 282 |
| 191. | Sample 2 - Convert DASD volume and all its data sets to SMS | 282 |
| 192. | Sample 3 - DFSMSdss data set copy and re-catalog | 283 |
| 193. | Sample 4 - DFSMSdss COPYDUMP JCL copy dump data set | 283 |
| 194. | Sample 5 - DFSMS copy ALLDATA including VOLID | 284 |
| 195. | Sample 6 - DFSMSdss DELETE | 284 |
| 196. | Sample 7 - DFSMSdss JCL to DUMP data sets | 285 |
| 197. | Sample 8 - DFSMSdss full volume physical dump | 285 |
| 198. | Sample 9 - DFSMSdss job to RELEASE unused space | 286 |
| 199. | Sample 10 - DFSMSdss logical data set restore | 286 |
| 200. | Sample 11 - DFSMSdss full volume RESTORE from tape | 286 |
| 201. | Sample 12 - DFSMSdss VSAM data set copy | 287 |
| 202. | Sample 13 - IDCAMS ALTER job to RENAME a data set | 287 |
| 203. | Sample 14 - AMASPZAP job to ZAP a load module | 288 |
| 204. | Sample 15 - AMBLIST job to LIST load module and CSECTs | 288 |
| 205. | Sample 16 - COBOL compile/LKED JCL | 289 |
| 206. | Sample 17 - IEBCOPY data set compress | 289 |
| 207. | Sample 18 - IDCAMS job to define an alias | 290 |
| 208. | Sample 19 - IEFBR14 define data set | 290 |
| 209. | Sample 20 - IDCAMS job to define a VSAM data set | 290 |
| 210. | Sample 21 - IDCAMS job to define a PAGESPACE data set | 291 |
| 211. | Sample 22 - IDCAMS DELETE and DEFINE VVDS | 291 |
| 212. | Sample 23 - IEFBR14 job to delete a data set | 291 |
| 213. | Sample 24 - IDCAMS delete VVR | 292 |
| 214. | Sample 25 - Produce error log report from Coupling Facility | 292 |
| 215. | Sample 26 - Produce error log report from SYS1.LOGREC data set | 293 |
| 216. | Sample 27 - ICKDSF job to perform DASD volume analysis | 293 |

| | | |
|------|--|-----|
| 217. | Sample 28 - ICKDSF REFORMAT can be used to change the VOLSER | 293 |
| 218. | Sample 29 - ICKDSF job to INITIALIZE a DASD volume | 294 |
| 219. | Sample 30 - ICKDSF job to INITIALIZE a DASD volume | 294 |
| 220. | Sample 31 - ICKDSF job to INSPECT a DASD device | 295 |
| 221. | Sample 32 - ICKDSF job to BUILD an indexed VTOC | 295 |
| 222. | Sample 33 - IEBCOPY partitioned data sets | 295 |
| 223. | Sample 34 - IEBGENER to copy data sets | 296 |
| 224. | Sample 35 - IEFBR14 job to allocate a partitioned data set | 296 |
| 225. | Sample 36 - IEHINITT to tape volume | 296 |
| 226. | Sample 37 - IEHLIST job to list a VTOC | 297 |
| 227. | Sample 38 - Assembler JCL (ASMA90) | 297 |
| 228. | Sample 39 - Sample LINKEDIT job | 297 |
| 229. | Sample 40 - IDCAMS job to review CACHE data for a volume | 298 |
| 230. | Sample 41 - IDCAMS LISTCAT | 298 |
| 231. | Sample 42 - Sample job to dump the SMF RACF records and report | 299 |
| 232. | Sample 43 - IDCAMS REPRO | 299 |
| 233. | Sample 44 - Stand-alone dump generation JCL | 300 |
| 234. | Sample 45 - Clear SMF data set (SYS1.MANx) | 300 |
| 235. | Sample 46 - Dump SMF SYS1.MANx data set | 301 |
| 236. | Sample 47 - IDCAMS user catalog DISCONNECT | 301 |

Tables

| | |
|---|-----|
| 1. Base Elements in OS/390 R7 | 263 |
| 2. Optional Elements in OS/390 R7 | 271 |
| 3. System programmer JCL toolbox | 281 |

Preface

This redbook is Volume 1 of a five-volume set that is designed to introduce the structure of an OS/390 and S/390 operating environment. The set will help you install, tailor, and configure an OS/390 operating system, and is intended for system programmers who are new to an OS/390 environment.

Volume 1 gives a broad understanding of the software and S/390 architecture and how it is used together with the OS/390 operating system.

Chapter 1 provides an introduction to the products and components that make up an OS/390 system.

Chapter 2 describes OS/390 storage concepts.

Chapter 3 describes S/390 hardware and I/O management which includes an introduction to HCD.

Chapter 4 describes the customization and installation products TSO/E and ISPF and job control language (JCL).

Chapter 5 describes the OS/390 delivery options and the download process using ServerPac option.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Paul Rogers is an OS/390 specialist at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on various aspects of OS/390. Before joining the ITSO 11 years ago, he worked in the IBM Installation Support Center (ISC) in Greenford, England as OS/390 and JES support for IBM EMEA.

Guillermo Capobianco is an IT Specialist in IBM Global Services PSS Argentina. He has five years of experience working with customers on MVS, MVS-related program products, and OS/390. He is currently leading a technical group providing on-site customer support for the OS/390 platform.

David Carey is a Senior IT Availability Specialist with the IBM Support Center in Sydney, Australia, where he provides defect and nondefect support for CICS, CICSplex/SM, MQSeries, and OS/390. David has 19 years of experience within the information technology industry, and was an MVS systems programmer for 12 years prior to joining IBM.

T. Nigel Davies is a Systems Specialist in IBM Global Services Product Support Services (PSS) in the United Kingdom. He has 10 years of IT experience in various roles, ranging from operations to PC and LAN support to mainframe systems programming. He joined IBM in 1997 with eight years of experience as a VM/VSE systems programmer, and since joining IBM has cross-trained in OS/390 systems skills. His areas of expertise include VM and VSE systems

programming, installation, and technical support, and more recently, OS/390 installation and support.

Luiz Fadel - IBM Brazil

Ken Hewitt is an IT Specialist in IBM Australia. He has over 10 years of experience working with S/390 customers in a range of roles from CE to System Engineer. His areas of expertise include I/O and OSA configuration.

Joao Natalino Oliveira

Joao Natalino de Oliveira is a certified I/T consulting specialist working for the S/390 in Brazil providing support for Brazil and Latin America. He has 24 years of experience in large systems including MVS-OS/390. His areas of expertise include performance and capacity planning, server consolidation and system programming. He has a bachelor degree in Math and Data Processing from Fundação Santo André Brazil.

Fabio Chaves Pita - IBM Brazil

Alvaro Salla has 30 years of experience in OS operating systems (since MVT). He has written several redbooks on S/390 subjects. Retired from IBM Brasil, he is now a consultant for IBM customers.

Valeria Sokal is an MVS system programmer at Banco do Brasil. She has 11 years of experience in the mainframe arena. Her areas of expertise include MVS, TSO/ISPF, SLR, and WLM.

Yoon Foh Tay is an IT Specialist with IBM Singapore PSS (S/390). He has six years of experience on the S/390 platform, providing on-site support to customers.

Hans-Juergen Timm is an Advisory Systems Engineer in IBM Global Services PSS Germany. He has 20 years of experience working with customers in the areas of MVS and OS/390, software and technical support, and planning and management. He also worked six years as an MVS Instructor in the IBM Education Centers in Mainz and Essen, Germany. His areas of expertise include implementation support for OS/390, Parallel Sysplex, UNIX System Services, and Batch Management.

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "IBM Redbooks evaluation" on page 325 to the fax number shown on the form.
- Use the online evaluation form found at <http://www.redbooks.ibm.com/>
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. Introduction to OS/390

OS/390 is an integrated enterprise server operating system. It incorporates into one product a leading-edge and open communications server, distributed data and file services, Parallel Sysplex system support, object-oriented programming, distributed computer environment (DCE), and open application interface. As such, it is uniquely suited to integrate today's heterogeneous and multi-vendor environments.

By incorporating the base operating system, it continues to build on the classic strengths of MVS: its reliability, continuous availability features, and security. This provides a scalable system that supports massive transaction volumes and large numbers of users with high performance, as well as advanced system and network management, security, and 24/7 availability.

1.1 Introduction to OS/390 fundamentals

OS/390 Release 1 was introduced as a replacement for MVS/ESA Version 5 Release 2. There have now been eight releases of OS/390, with a new release planned every six months. This book was based on OS/390 Release 7; Version 2. Release 8 was made generally available in September 1999.

Before OS/390, installations used an MVS operating system that consisted of:

- The Base Control Program (BCP)

The Base Control Program (BCP) provides essential operating system services. The BCP includes the I/O configuration program (IOCP) as well as the function that starts the OS/390 UNIX System Services address space. This latter function was called OpenEdition System Services and was a nonexclusive base element of OS/390 R1, was an exclusive element of OS/390 R2, was integrated into the BCP element in OS/390 R3, and in OS/390 R6 remained as a part of the BCP but its name was changed to OS/390 UNIX System Services.

- DFSMSdfp
- JES2 or JES3
- TSO/E and ISPF
- A collection of other software products that applications required

With OS/390, IBM integrated these products and offered a single product. With OS/390, you may order new levels of some products but not of others; instead, you order and install an entire set of products integrated into one functionally rich operating system.

Once positioned on OS/390, you can receive new functions and levels approximately every six months. This predictable release cycle helps reduce planning time. Because each new level is comprehensively tested, the quality of the operating system is improved. Once your initial migration is complete, you can expect simplified ordering, planning, and installing, freeing you to increase the value of your computing environment to your business and deliver better service to your end users.

OS/390: S/390 Server Operating System

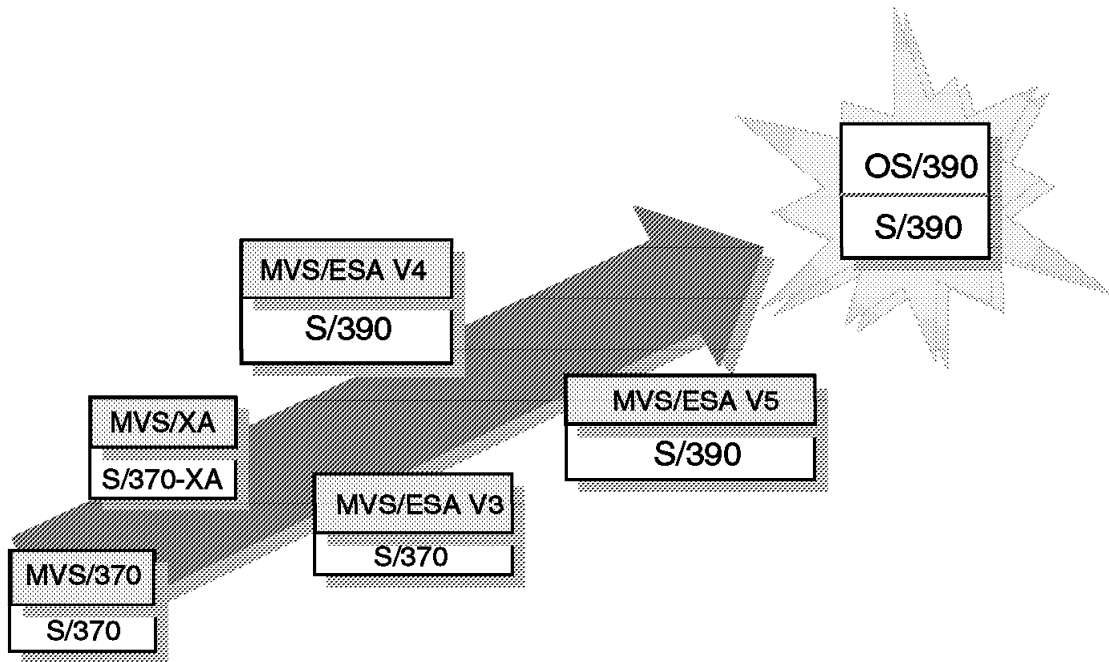


Figure 1. OS/390: S/390 Server Operating System

1.1.1 OS/390 evolution

The enterprise systems hardware was transformed by the introduction of the OS/390 Parallel Servers. Based on the complementary metal oxide semiconductor (CMOS) technology, these parallel systems are smaller in size and larger in capacity than their predecessors and deliver performance at a lower cost.

Now, the transformation continues with the system software for S/390. OS/390 is the S/390 server operating system. OS/390 extends S/390's architecture to provide the enterprise-wide client/server infrastructure and tools that businesses need for fast, flexible deployment of new applications.

With the introduction of the S/390 Parallel Servers in 1994, the mainframe was revived. The CMOS technology offers mainframe computing power at a lower cost. The S/390 CMOS machines can be connected with other S/390 CMOS machines or traditional ES/9000 machines to form a Parallel Sysplex. The sysplex offers high availability and the option to add capacity in small increments. The S/390 CMOS technology is a modern solution.

The transformation of the mainframe hardware together with business requirements for Information Technology (IT) are driving forces behind a change in the operating system software for the S/390 mainframes. In 1996, the OS/390 system was introduced as the S/390 server operating system.

Hardware / Software Requirements

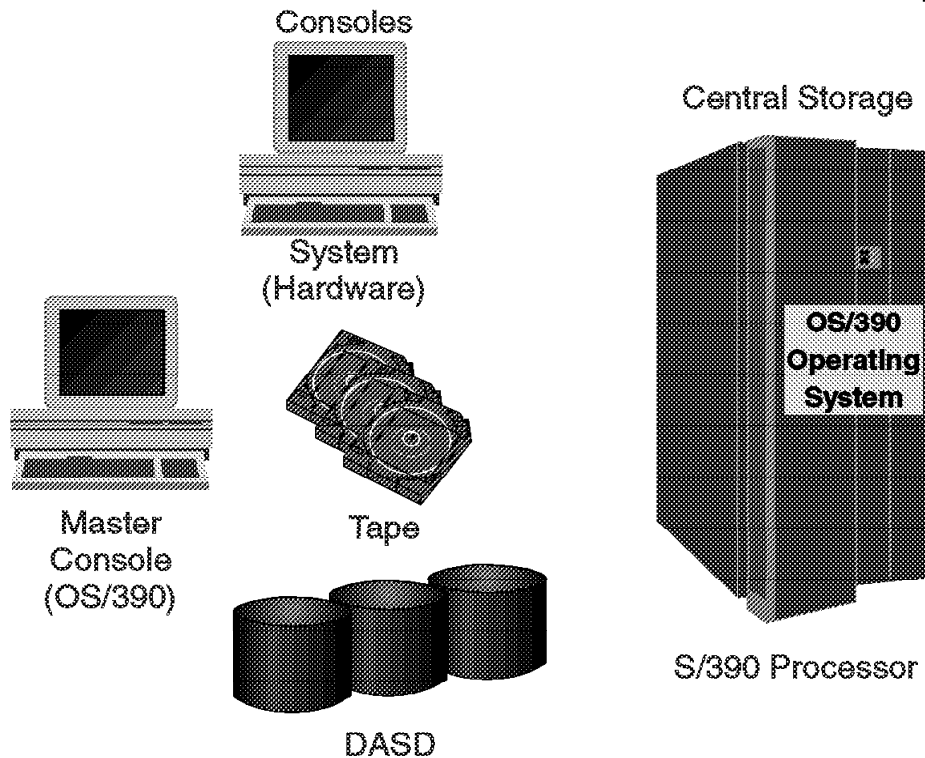


Figure 2. Hardware/Software Requirements

1.1.2 OS/390 software and hardware

The base OS/390 operating systems executes in a processor and resides in the processor storage during execution. The OS/390 operating system is commonly referred to as the system software.

The hardware consists of the processors and other devices such as a direct access storage device (DASD), tape, and consoles. Tape and DASD are used for system functions and by user programs that execute in an OS/390 environment. When you order OS/390, you receive your order on tape cartridges. When you install the system from tape, the system code is then stored on DASD volumes. Once the system is customized and ready for operation, system consoles are required to start and operate the OS/390 system.

Not shown at this time in the visual are the control units that connect the CPU (processor) to the other tape, DASD, and console devices.

The main concepts shown here are:

- Software** The OS/390 operating system consists of load modules and is often called executable code. These load modules are placed onto DASD volumes into load libraries during a system install process.
- Hardware** The system hardware consists of all the devices, controllers, and processors that make up an OS/390 complex.
- Devices** Shown in the visual are the tape, DASD, and console devices. There are many other types of devices that will be discussed later in this document.

Storage

Central storage, often called real or main storage, is where the OS/390 operating system executes. Also, all user programs share the storage of the processor with the operating system.

OS/390 Elements and Features



Base Elements:

- ★ **MVS/ESA**
- ★ **DFSMSdfp**
- ★ **JES2**
- ★ **TSO/E**
- ★ **ISPF**
- ★ **ACF/VTAM**
- ★ **GDDM**
- ★ **BookManager READ**
- ★ **UNIX System Services**
- ★ **SMP/E**
- ★ **SOMobjects for MVS RTL**
- ★ **GDDM/MVS**

Optional Features:

- ★ **JES3**
- ★ **DFSMSdss**
- ★ **DFSMSHsm**
- ★ **DFSMSrmm**
- ★ **RMF**
- ★ **RACF**
- ★ **IBM C/C++ compiler**
- ★ **SDSF**
- ★ **DFSORT**
- ★ **IBM TCP/IP FOR MVS**
- ★ **SOMobjects for MVS ADE**
- ★ **GDDM-PGF**

Figure 3. OS/390 Elements and Features

1.1.3 OS/390 elements and features

The OS/390 system consists of base elements that deliver essential operating functions. In addition to the services provided by MVS/ESA, this means such functions as communications support, online access, host graphics, and online viewing of publications.

In addition to the base, OS/390 has optional features that are closely related to the base features. There are two types of optional features:

- One type of feature is always shipped with the OS/390 system whether they are ordered or not. These features support dynamic enablement, which allows you to dynamically enable and disable them.
 - If such a feature is ordered, it is shipped enabled for use.
 - If such a feature is not ordered, it is shipped disabled. It can later be enabled.
- A second type of optional feature is not shipped automatically. These features must be ordered specifically.

The idea of the OS/390 system is to have elements and features instead of program products. This concept might be more easily explained by saying that OS/390 consists of a collection of functions that are called base elements and optional elements. The optional elements (features) are either integrated or nonintegrated. It is important to note that these optional features, both integrated and nonintegrated, are also tested as part of the integration of the entire system. The intention of this visual is to explain the difference between these terms. It is not the intention to discuss which products are included in OS/390 and which are not.

- Shipped as part of the OS/390 system is the base operating system and the products/features that were part of MVS/ESA SP V5.2.2, for example, UNIX System Services, SOMobjects, and LAN services. In addition to these features, products such as VTAM, TSO/E, ISPF, GDDM ., and BookManager READ/MVS, which provide essential operating system functions are included in the base and called base elements. The list on the visual is not a complete list. More details will be provided later in the course. Some of the base elements can be dynamically enabled and disabled, for example TCP/IP and DFSMS/NFS. The reason for this is that a customer may choose to use a vendor product for TCP/IP and NFS instead of IBM's products.
- In addition to the OS/390 base, there is a set of optional features. Note that there are two types of optional features: one type is always shipped, and the other must be specifically ordered.
 - The features that support dynamic enablement are always shipped. Examples are JES3, DFSMSdss, and DFSMSHsm. If these features are ordered as part of the OS/390 system order, they will be shipped as enabled in the system. If they are not ordered, they are shipped as disabled. Later on, you can use them by letting IBM know and by dynamically enabling them through a SYS1.PARMLIB member.
 - The other type of features are the optional features equivalent to optional program products. Examples are RACF, RMF, C/C++ compiler, and so on.

Some of the optional products will still be available as separate orderable products for customers that are using MVS/ESA. However, it is IBM's intention to provide new functions only within the OS/390 elements and features. Future releases of the OS/390 system will contain more elements and features as more program products are included in the solution.

Note: The list of elements and features on the visual is not a complete list of what is included with the OS/390 system.

Additional Information — There are two classifications of elements in the OS/390 system: exclusive and nonexclusive.

- Exclusive elements:
 - The functional level of an element or feature that can be ordered only as part of the OS/390 package, and is not available as an independent element or feature anywhere else.
- Nonexclusive elements:
 - Those elements or features included in the OS/390 package that are also orderable as independent products, at the same functional level, from the MVS product set.

OS/390 Products Requiring System Programmer Customization



- ★ RACF - (Security Server)
- ★ DFSMS
- ★ ADSM
- ★ eNetwork Communications Server
 - ▶ TCP/IP and VTAM/SNA
- ★ SMP/E
- ★ RMF

Figure 4. Products Requiring System Programmer Customization

1.2 OS/390 products requiring customization

The products shown on the visual need to be installed and customized for use in an OS/390 operating system.

When ordering products, be sure you include the following steps when planning your pre-installation activities:

- Obtain and install any required program temporary fixes (PTFs) or updated versions of the operating system.
- Call the IBM Software Support Center to obtain the preventive service planning (PSP) upgrade. This provides the most current information on PTFs for RACF. Have RETAIN checked again just before testing RACF. Information for requesting the PSP upgrade can be found in the program directory. Although the program directory contains a list of the required PTFs, the most current information is available from the support center.
- Verify that your installation's programs will continue to run, and, if necessary, make changes to ensure compatibility with the new release.

OS/390 SecureWay Security Server (RACF Component)



Resource Access Control Facility (RACF)

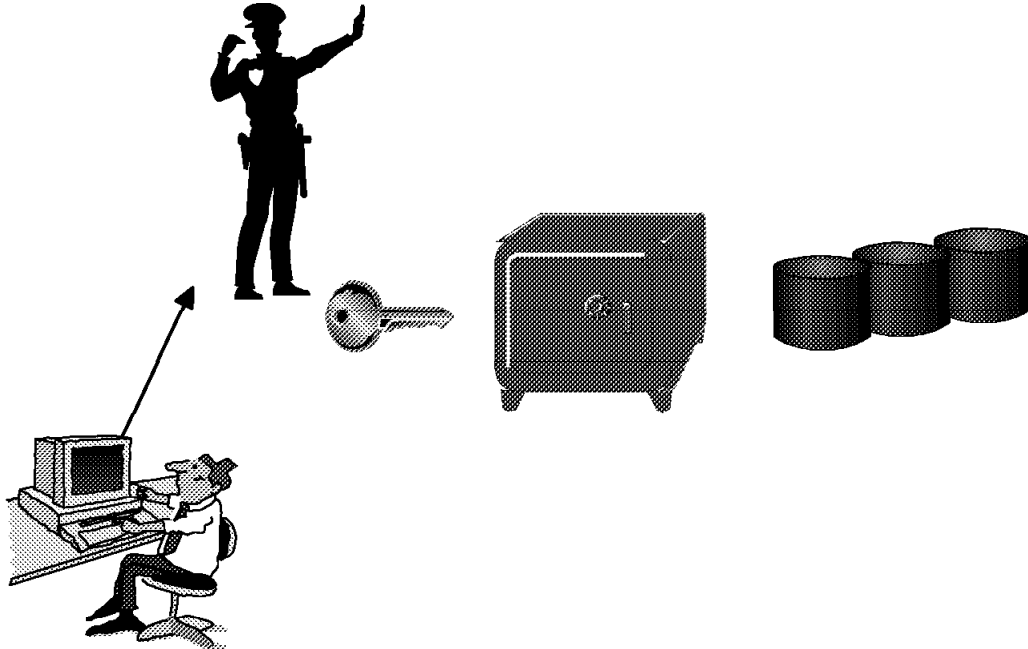


Figure 5. RACF - Resource Access Control Facility

1.2.1 Resource Access Control Facility (RACF)

The OS/390 Security Server (RACF component) is the IBM security product. The RACF component of the OS/390 Security Server works together with the existing system features of OS/390 to provide improved data security for an installation and provides Year 2000 support. If this product is the one to be installed in your environment, then RACF customization must be done.

RACF helps meet the need for security by providing:

- Flexible control of access to protected resources
- Protection of installation-defined resources
- Ability to store information for other products
- Choice of centralized or decentralized control of profiles
- An ISPF panel interface
- Transparency to end users
- Exits for installation-written routines

In order for RACF to meet the specific requirements of your installation, you can customize function to take advantage of new support after the product is installed. For example, you can tailor RACF through the use of installation exit routines, class descriptor table (CDT) support, or options to improve performance.

DFSMS

Data Facility Storage Management Subsystem

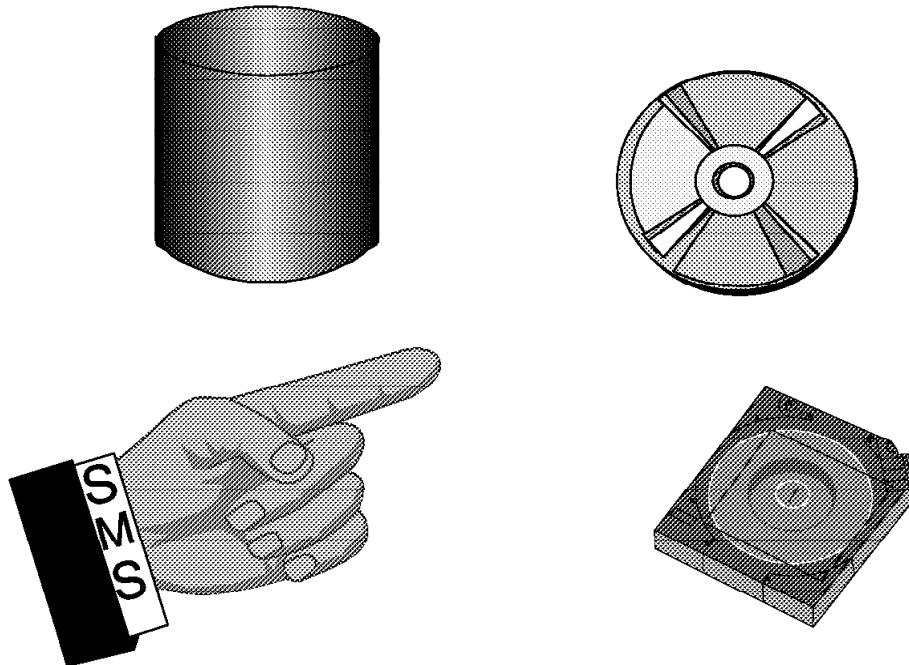


Figure 6. Data Facility Storage Management Subsystem

1.2.2 Data Facility Storage Management Subsystem (DFSMS)

DFSMS/MVS and MVS comprise the base MVS operating system, where DFSMS/MVS performs the essential data, storage, program, and device management functions of the system.

DFSMS/MVS is the central component of both system-managed and non-system-managed storage environments. DFSMS/MVS, MVS, and ESA/370 or ESA/390 hardware exploit the usability and function available with MVS. MVS supports both 24-bit and 31-bit addressing used by components of DFSMS/MVS. Many DFSMS/MVS components have modules or data in extended virtual storage above 16 MB, leaving more space below the 16 MB line for user applications.

The DFSMS environment consists of a set of IBM hardware and software products that together provide a system-managed storage solution for MVS installations. DFSMS/MVS is an integral part of this environment.

The components of DFSMS/MVS automate and centralize storage management based on installation-defined policies for availability, performance, space, and security. The Interactive Storage Management Facility (ISMF) provides the user interface for defining and maintaining these policies and the Storage Management Subsystem (SMS) governs these policies for the system.

In this environment, the Resource Access Control Facility (RACF) and Data Facility Sort (DFSORT) complement the functions of the base operating system; RACF provides resource security functions, and DFSORT adds the capability for faster and more efficient sorting, merging, copying, reporting and analyzing of business information.

ADSM

ADSTAR Distributed Storage Management



ADSM
Storage Pools

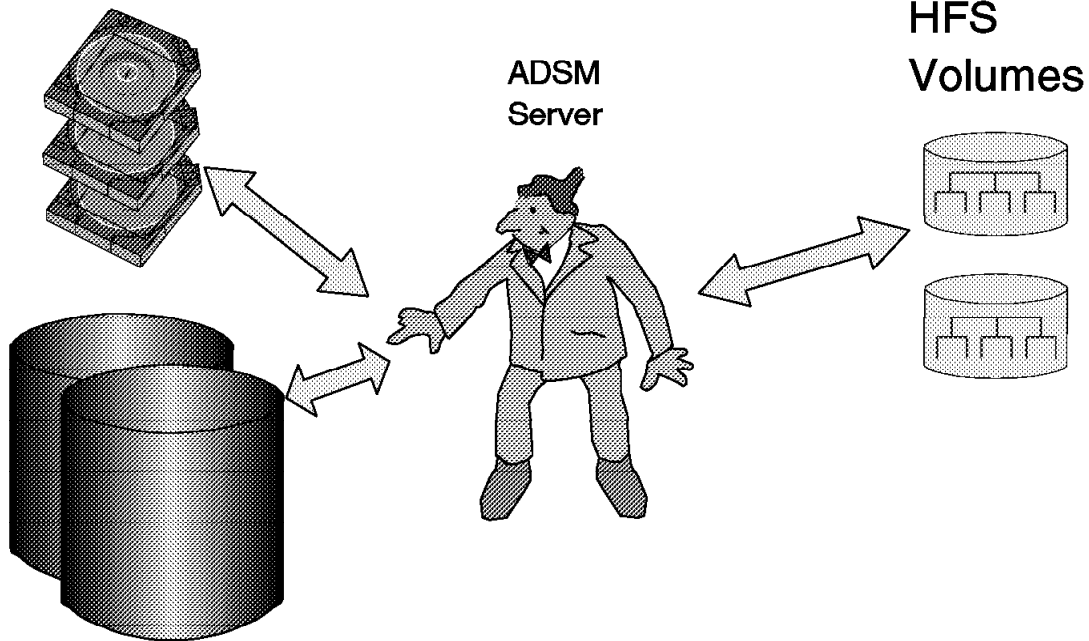


Figure 7. ADSTAR Distributed Storage Management

1.2.3 ADSTAR Distributed Storage Management (ADSM)

ADSTAR Distributed Storage Manager (ADSM) is an enterprise-wide storage management application for the network. It provides automated storage management services to multivendor workstations, personal computers, and local area network (LAN) file servers.

This product is useful for backing up files for workstations, personal computers, LAN file servers, and OS/390 UNIX System Services HFS files.

TCP/IP

Transmission Control Protocol/Internet Protocol

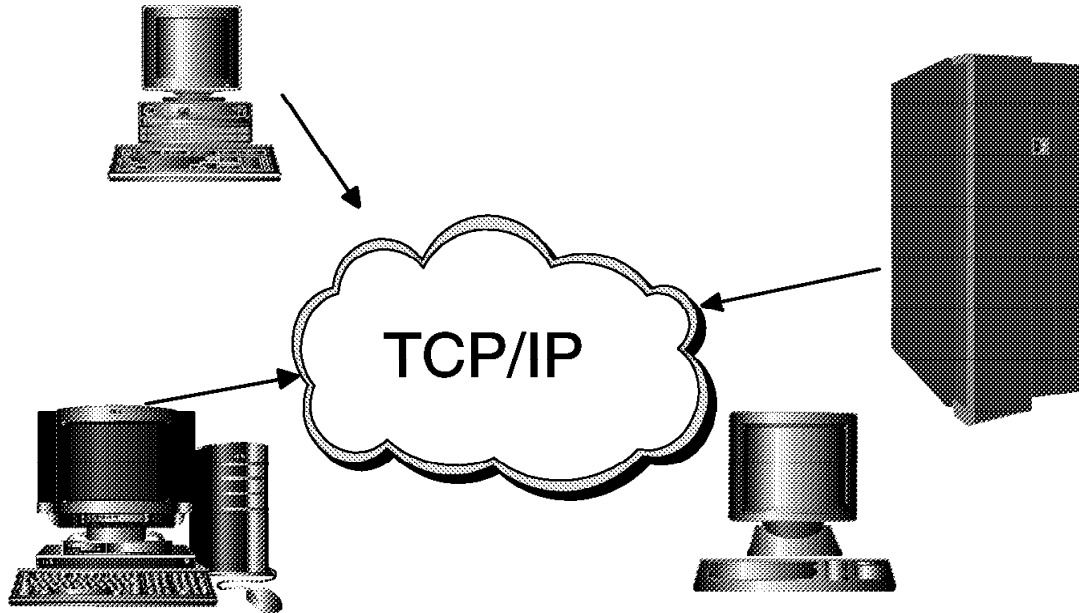


Figure 8. Transmission Control Protocol/Internet Protocol

1.2.4 Transmission Control Protocol/Internet Protocol (TCP/IP)

TCP/IP is a set of protocols and applications that allow you to perform certain computer functions in a similar manner independent of the types of computers or networks being used. When you use TCP/IP, you are using a network of computers to communicate with other users, share data with each other, and share the processing resources of the computers connected to the TCP/IP network.

A computer network is a group of computer nodes electronically connected by some communication medium. Each node has the hardware and the programs necessary to communicate with other computer nodes across this communication medium. The node can be a PC, workstation, microcomputer, departmental computer, or large computer system. The size of the computer is not important. The ability to communicate with other nodes is important.

Computer networks allow you to share the data and computing resources of many computers. Applications, such as departmental file servers, rely on networking as a way to share data and programs.

Many forms of communication media are available today. Each is designed take advantage of the environment in which it operates. Communication media consist of a combination of the physical network used to connect to computer nodes and the language, or protocol, they use to communicate with each other.

SMP/E

System Modification Program Extended



Software Installation

Maintenance

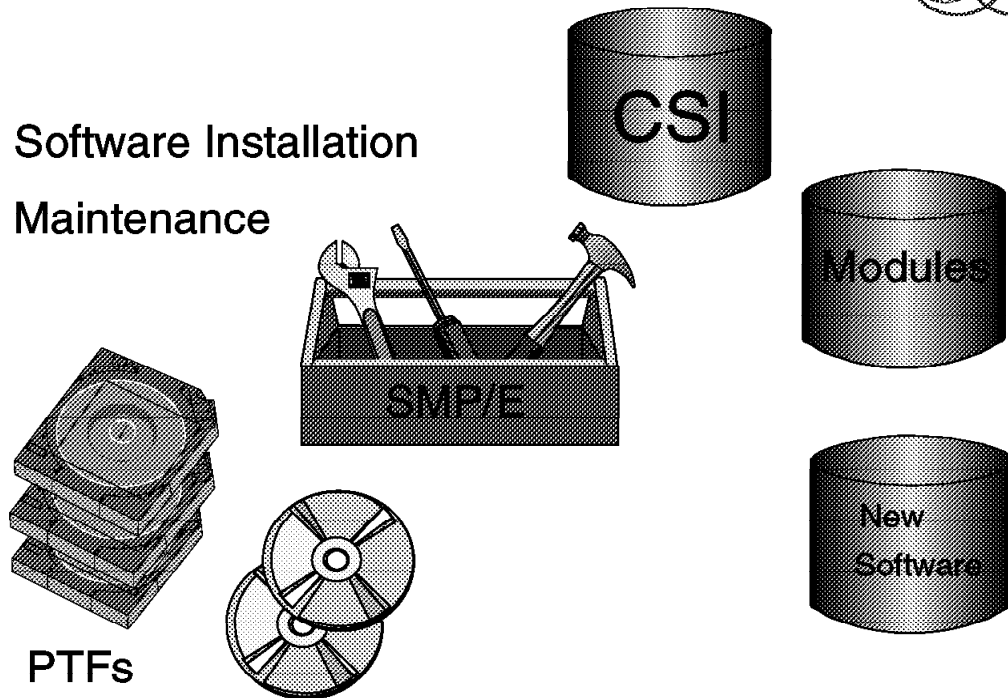


Figure 9. System Modification Program Extended (SMP/E)

1.2.5 System Modification Program Extended (SMP/E)

System Modification Program Extended (SMP/E) is a tool designed to manage the installation of software products on your MVS system, and to track the modifications you make to those products. Usually, it is the system programmer's responsibility to ensure that all software products and the modifications are properly installed on the system. The system programmer also has to ensure that all products are installed at the proper level so all elements of the system can work together. At first, that might not sound too difficult, but as the complexity of the software configuration increases, so does the task of monitoring all the elements of the system.

An OS/390 system may appear to be one big block of code that drives the CPU. Actually, OS/390 is a complex system comprising many different smaller blocks of code. Each of those smaller blocks of code perform a specific function within the system. Each system function is composed of one or more load modules. In an OS/390 environment, a load module represents the basic unit of machine-readable executable code. Load modules are created by combining one or more object modules and processing them with a link-edit utility. The link-editing of modules is a process that resolves external references and addresses. The functions on a system, therefore, are one or more object modules that have been combined and link-edited.

Over time, you may need to change some of the elements of your system. These changes may be necessary to improve the usability or reliability of a product. You may want to add some new functions to your system, upgrade some of the elements of your system, or modify some elements for a variety of reasons. In all cases, you are making system modifications. In SMP/E we refer to these system modifications as SYSMODs.

RMF Resource Measurement Facility

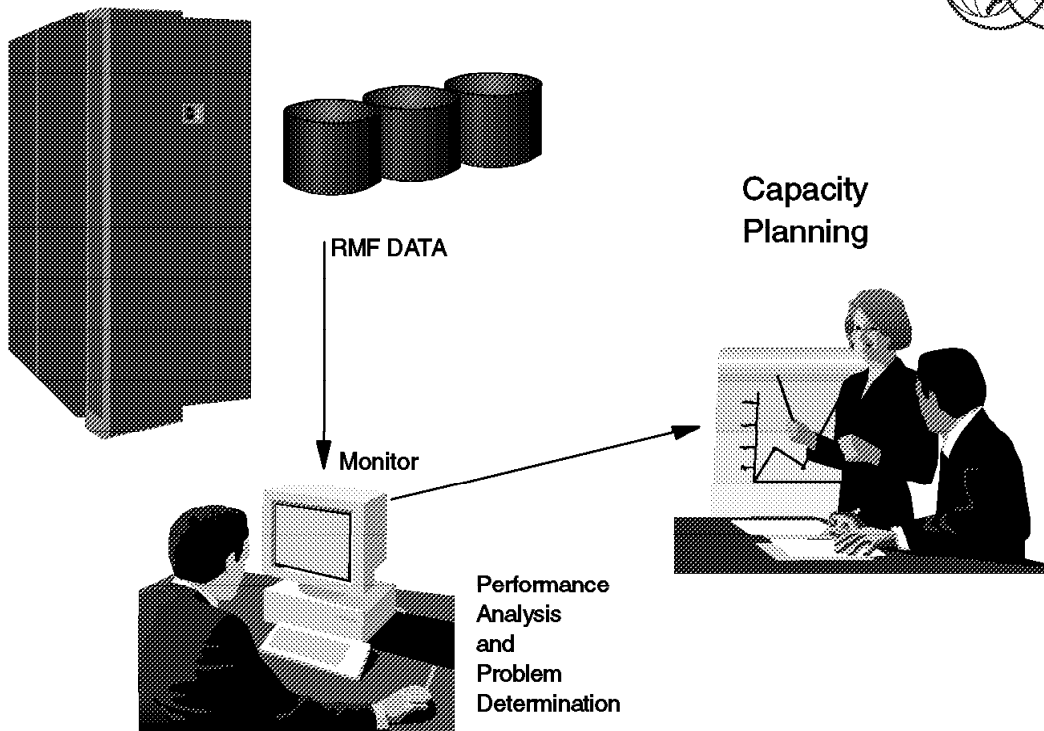


Figure 10. Resource Management Facility (RMF)

1.2.6 Resource Management Facility (RMF)

Many different activities are required to keep your OS/390 running smoothly, and to provide the best service on the basis of the available resources and workload requirements. The console operator, the service administrator, the system programmer, or the performance analyst will do these tasks. RMF is the tool that helps each of these people do the job effectively.

RMF gathers data using three monitors:

- Short-term data collection with Monitor III
- Snapshot monitoring with Monitor II
- Long-term data gathering with Monitor I

Data is gathered for a specific cycle time, and consolidated data records are written at a specific interval time. The default value for data gathering is one second and for data recording 30 minutes. You can select these options according to your requirements and change them whenever the need arises.

Monitor I collects long-term data about system workload and resource utilization, and covers all hardware and software components of your system: processor, I/O device and storage activities and utilization, as well as resource consumption, activity and performance of groups of address spaces.

OS/390 Components Requiring System Programmer Customization



- ★ SMF
- ★ VLF
- ★ TSO/E
- ★ WLM
- ★ OS/390 UNIX System Services

Figure 11. Components Requiring System Programmer Customization

1.3 OS/390 components requiring customization

The components shown on the visual are part of OS/390 and they require customization. These components are part of the base MVS system and key components for running the OS/390 operating system.

SMF System Management Facility

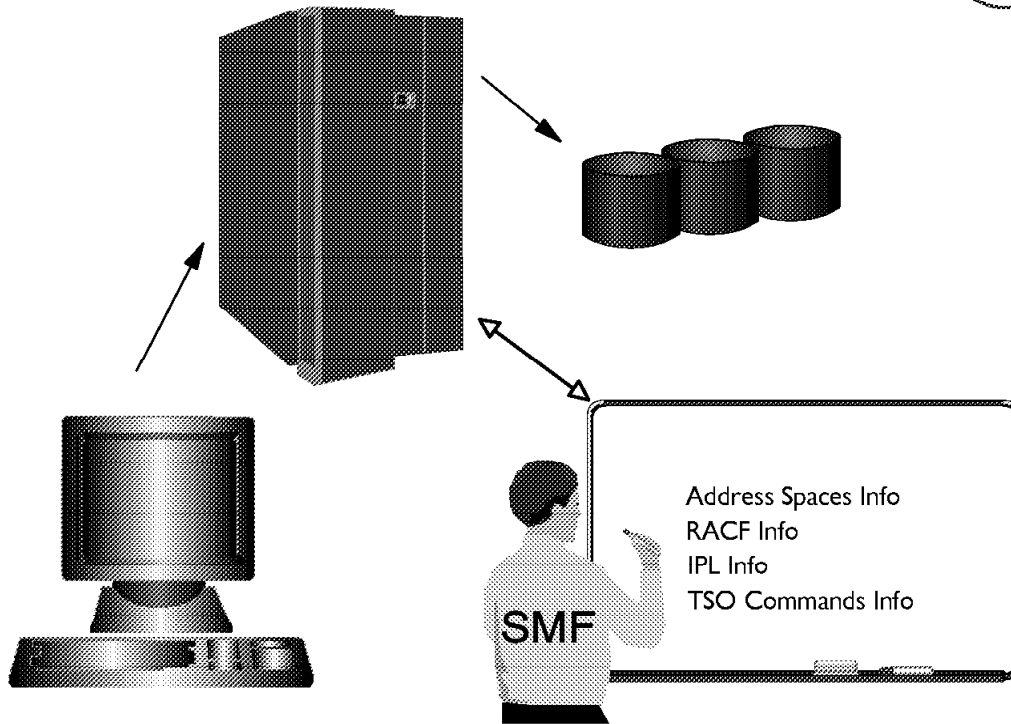


Figure 12. System Management Facility (SMF)

1.3.1 System Management Facility (SMF)

System management facilities (SMF) collects and records system and job-related information that your installation can use in:

- Billing users.
- Reporting reliability.
- Analyzing the configuration.
- Scheduling jobs.
- Summarizing direct access volume activity.
- Evaluating data set activity.
- Profiling system resource use.
- Maintaining system security.

SMF formats the information that it gathers into system-related records (or job-related records). System-related SMF records include information about the configuration, paging activity, and workload. Job-related records include information on the CPU time, SYSOUT activity, and data set activity of each job step, job, APPC/MVS transaction program, and TSO/E session.

An installation can provide its own routines as part of SMF. These routines will receive control either at a particular point as a job moves through the system, or when a specific event occurs. For example, an installation-written routine can receive control when the CPU time limit for a job expires

or when an initiator selects the job for processing. The routine can collect additional information, or enforce installation standards.

VLF

Virtual Lookaside Facility

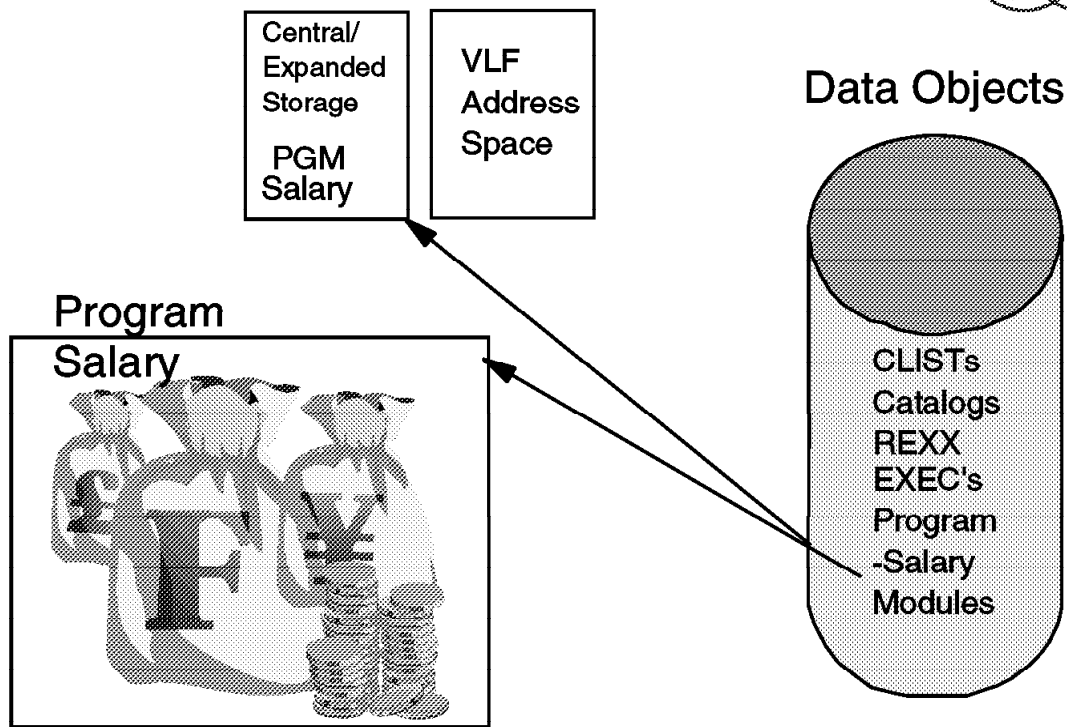


Figure 13. Virtual Lookaside Facility (VLF)

1.3.2 Virtual Lookaside Facility (VLF)

Virtual lookaside facility (VLF) is a set of services that can improve the response time of applications that must retrieve a set of data for many users. VLF creates and manages a data space to store an application's most frequently used data. When the application makes a request for data, VLF checks its data space to see if the data is there. If the data is present, VLF can rapidly retrieve it without requesting I/O to DASD.

To take advantage of VLF, an application must identify the data it needs to perform its task. The data is known as a data object. Data objects should be small to moderate in size, named according to the VLF naming convention, and associated with an installation-defined class of data objects.

Certain IBM products or components such as LLA, TSO/E, CAS, and RACF use VLF as an alternate way to access data. Since VLF uses virtual storage for its data spaces, there are performance considerations each installation must weigh when planning for the resources required by VLF.

Note: VLF is intended for use with major applications. Because VLF runs as a started task that the operator can stop or cancel, it cannot take the place of any existing means of accessing data on DASD. Any application that uses VLF must also be able to run without it.

1.3.2.1 Using VLF with LLA (Library Lookaside)

Library lookaside (LLA) is a started task system address space that improves the system's performance by reducing the contention for disk volumes, the searching of library directories, and the loading of programs. Directory entries for the primary system library, SYS1.LINKLIB, program libraries concatenated to it in SYS1.PARMLIB(LNKLSTxx), and additional production libraries named in SYS1.PARMLIB(CSVLLAxx) are read into the private area of the LLA address space during its initialization. Subsequent searches for programs in these libraries will begin with the directories in LLA, and not in the data sets on DASD. The most active modules from LLA-managed libraries are staged into the DCSVLLA data space managed by VLF.

You will obtain the most benefit from LLA when you have both LLA and VLF functioning. You should plan to use both.

TSO/E Time Sharing Option/Extended

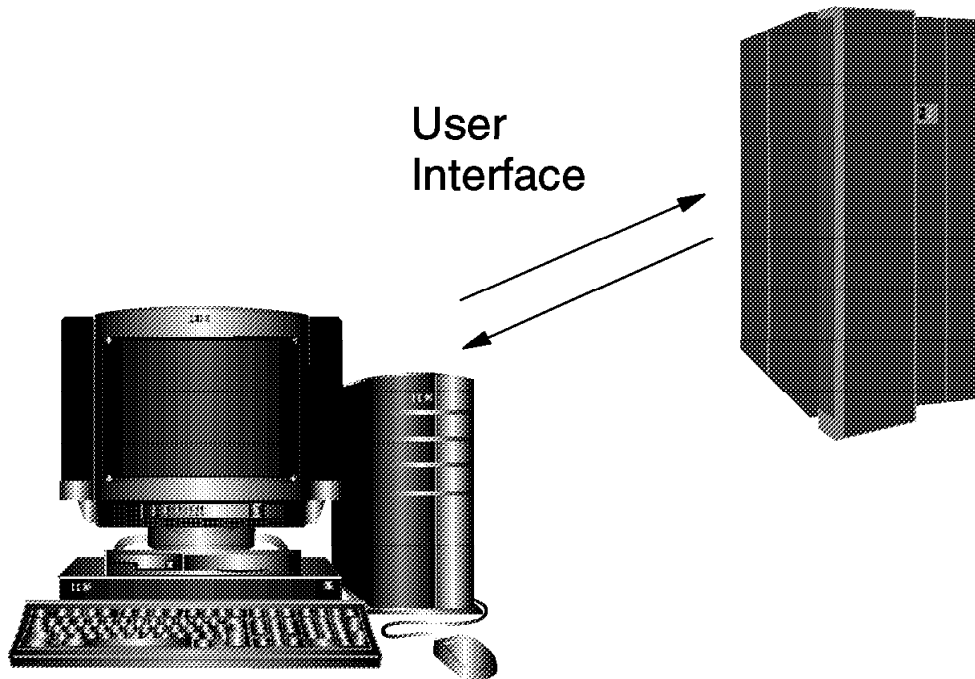


Figure 14. Time Sharing Option/Extended (TSO/E)

1.3.3 Time Sharing Option/Extended (TSO/E)

TSO/E is a base element of OS/390. TSO/E allows users to interactively share computer time and resources. In general, TSO/E makes it easier for people with all levels of experience to interact with the MVS system.

Before OS/390, TSO Extensions (TSO/E) was a licensed program for the MVS and MVS/ESA System Products, and it was an extension of the Time Sharing Option (TSO) of former MVS systems.

TSO/E has advantages for a wide range of computer users. TSO/E users include system programmers, application programmers, information center administrators, information center users, TSO/E administrators, and others who access applications that run under TSO/E.

WLM Workload Manager

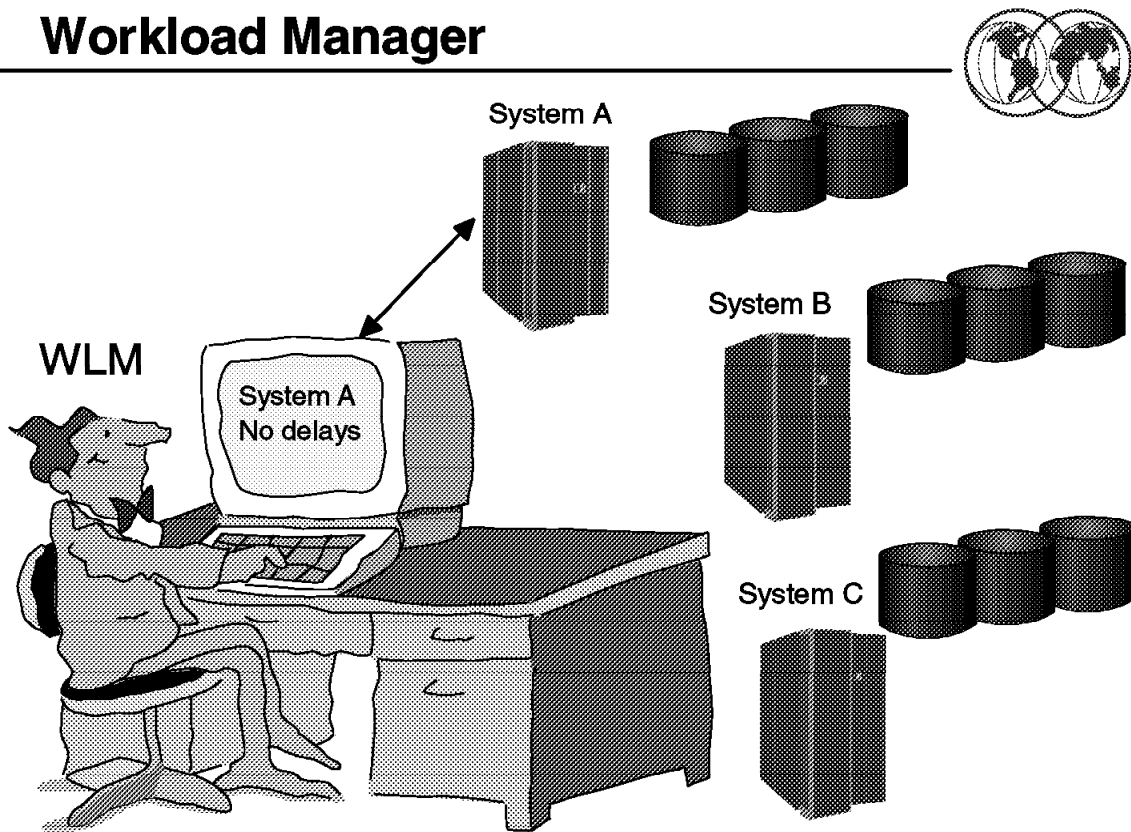


Figure 15. Workload Manager (WLM)

1.3.4 Workload Manager (WLM)

Before the introduction of MVS Workload Manager, MVS required you to translate your data processing goals from high-level objectives about what work needs to be done into the extremely technical terms that the system can understand. This translation requires high skill-level staff, and can be protracted, error-prone, and eventually in conflict with the original business goals. Multisystem, sysplex, parallel processing, and data sharing environments add to the complexity.

MVS Workload Manager provides a solution for managing workload distribution, workload balancing, and distributing resources to competing workloads. MVS Workload Manager is the combined cooperation of various subsystems (CICS, IMS/ESA, JES, APPC, TSO/E, OS/390 UNIX System Services, DDF, DB2, SOM, LSFM, and Internet Connection Server) with the MVS Workload Manager (WLM) component.

UNIX System Services

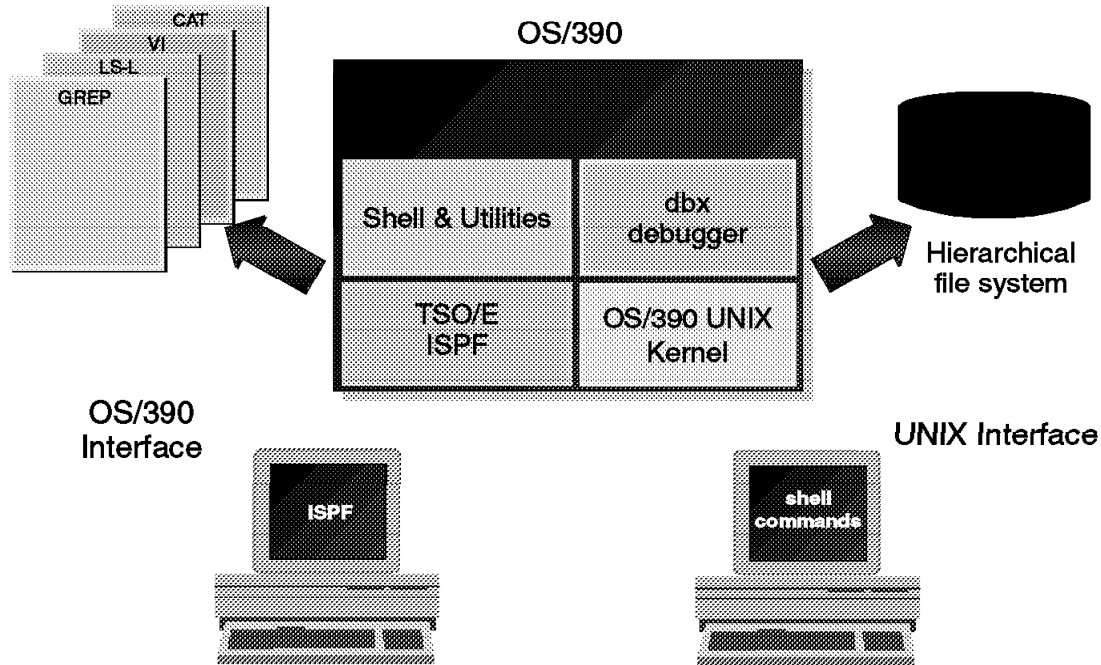


Figure 16. UNIX System Services

1.3.5 UNIX System Services

Beginning with OS/390 Release 3, UNIX System Services has been merged with the BCP, and is now part of the BCP FMID. In addition, the OMVS address space is started automatically.

BPXOINIT is the started procedure that runs the initialization process. The OMVS address space is now started automatically at IPL by means of the OMVS= statement in the IEASYSxx parmlib member.

OS/390 UNIX interacts with the following elements and features of OS/390:

- C/C++ Compiler, to compile programs
- Language Environment, to execute the shell and utilities or any other
- XPG4-compliant shell application
- Data Facility Storage Management Subsystem/MVS (DFSMS/MVS*)
- OS/390 Security Server
- Resource Measurement Facility (RMF)
- System Display and Search Facility (SDSF)
- Time Sharing Option Extensions (TSO/E)
- eNetwork Communications Server - TCP/IP Services (called SecureWay Communications Server with Release 8)
- ISPF, to use the dialogs for OEDIT, or ISPF/PDF for the ISPF shell

- BookManager* READ/MVS, to use the OHELP online help facility

OS/390 Contents

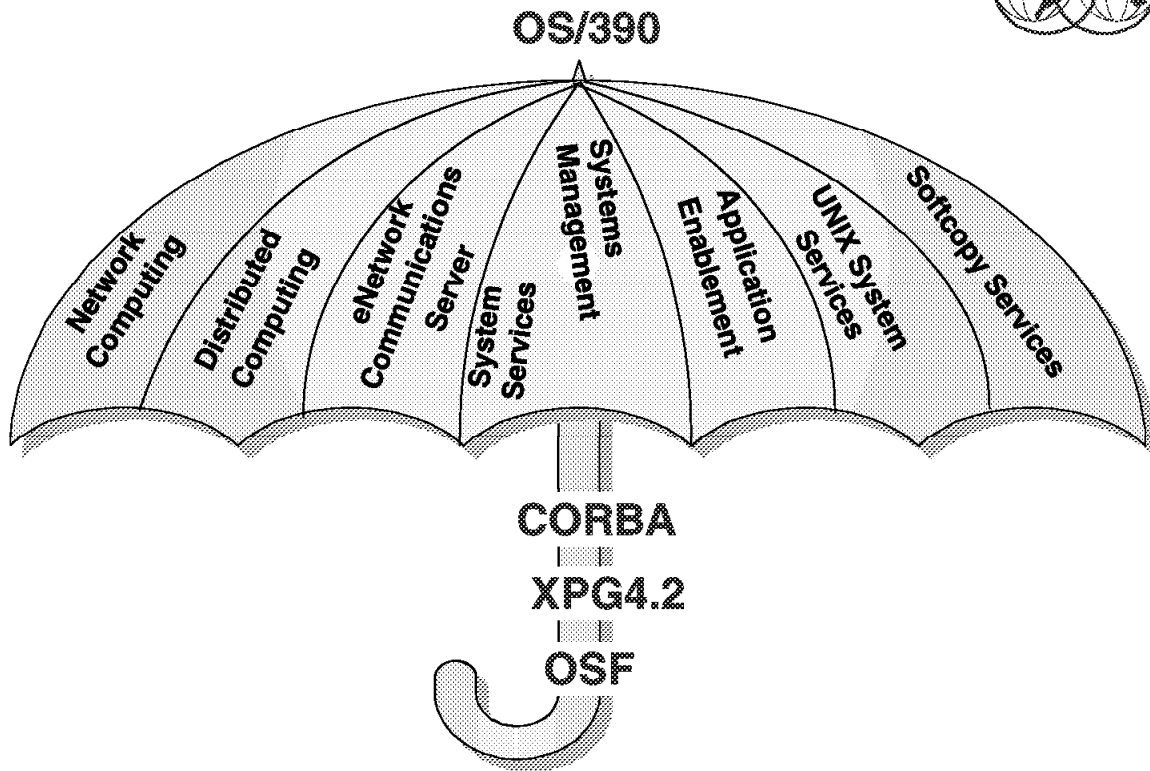


Figure 17. OS/390 Contents

1.3.6 OS/390 contents

OS/390 is based upon the MVS/ESA SP V5.2.2 product, and the latest versions of associated products. The OS/390 system provides solutions for the following major areas:

- **LAN Services:** Provides support for S/390 to be data and print servers in a Local Area Network (LAN) environment, as well as a focal point for LAN administration enabling LAN workstation users to store and share data and applications in a central location on the S/390.
- **Distributed Computing:** Support for distributed applications using industry solutions such as Distributed Computing Environment (DCE), Distributed File System (DFS), and Network File System (NFS).
- **eNetwork Communications Server:** (Also known as CS for OS/390 and SecureWay Communications Server) provides connectivity to a broad set of users and vendor platforms opening OS/390 for networking applications (called SecureWay Communications Server with Release 8).
- **System Services:** Provide the classic strengths of MVS, rock-solid reliability and availability, support for high-transaction workloads and many users with optimum performance.
- **Systems Management:** Provide a window to enterprise-wide systems management.
- **Application Enablement:** Support for the new object technology and rapid development of applications, improving time-to-market for new business function.
- **UNIX System Services:** Support for open standards such as Posix and XPG4.2 provides opportunities for more applications on the S/390 platform.
- **Softcopy Services:** Improves productivity in systems installation and management.
- **Network Computing Services:** Supports Secure Access to the Internet with Domino Go Webserver.
- **NetQuestion:** Provides a powerful, full-text indexing and search server. It supports high-speed searching of OS/390 Web sites, as well as documents stored on the OS/390 server.

Purpose — Introduce the contents and solutions provided by the OS/390 system.

Details — The OS/390 system is based upon the MVS/ESA SP V5.2.2 system and associated products. However, OS/390 contains functional changes in many of the base elements and features that are exclusive to OS/390.

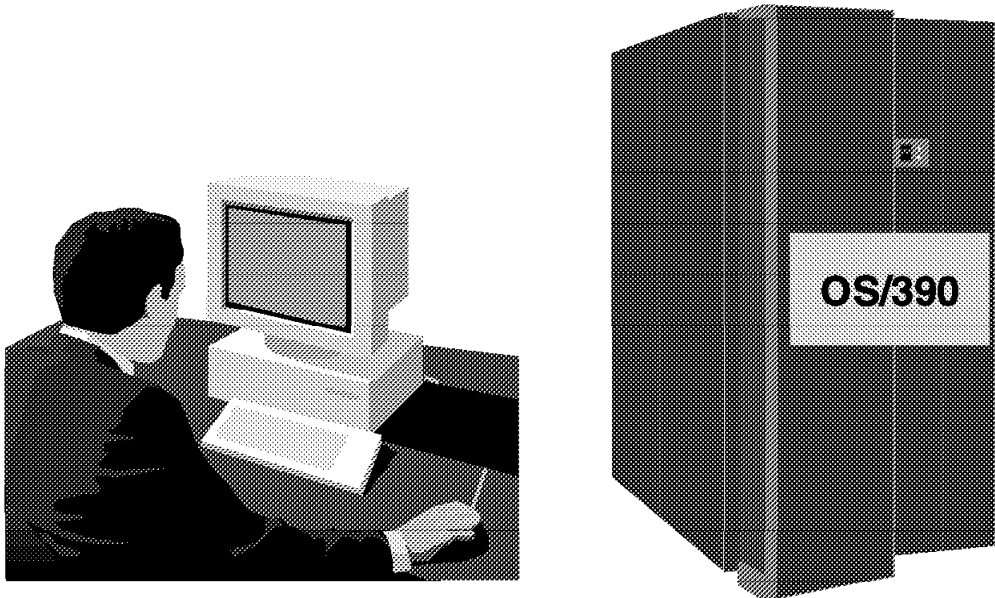
The Installation and Planning for Migration units present information on how to install and migrate to OS/390.

Use this visual to show that the OS/390 system can be looked upon as an umbrella solution for all the operating system functions in a S/390 environment. It supports several important industry standards: CORBA for objects, OSF for open distributed applications, and XPG4.2 for open systems support. It is a complete server system providing solutions for new technologies such as objects, client/server, and LAN just to mention a few. Each of these areas will be briefly presented on the following visuals.

The text shows an overview of the OS/390 system as well as the structure of this course. This course will also include some information on how to implement the server solutions.

Transition Statement — Before proceeding with an overview of each of the server areas, the next two visuals define the base elements and features that make up OS/390.

OS/390 Operating System



System Programmer

Figure 18. OS/390 Operating System

1.4 Role of a system programmer

The role of the system programmer is to install, customize, and maintain the operating system. The OS/390 operating system runs on various hardware configurations. A system programmer must also define the hardware I/O configuration resources that are to be available to the OS/390 operating system.

The hardware used can be either IBM or other manufacturers' machines. The hardware can be used in two modes:

- Basic mode** A central processor mode that does not use logical partitioning. With one central processor, one copy of the OS/390 operating system runs in the machine.
- LPAR mode** Logically partitioned (LPAR) mode, which is a central processor complex (CPC) power-on reset mode that enables use of the PR/SM feature and allows an operator to allocate CPC hardware resources (including central processor central storage, expanded storage, and channel paths) among logical partitions. OS/390 as the operating system runs in each LPAR in the machine.

As an OS/390 system programmer, you must be aware of the following:

- Storage concepts
- Device I/O configurations
- Processor configurations
- Console definitions

- System libraries where the software is placed
- System data sets and their placement
- Customization parameters that are used to define your OS/390 configuration

OS/390 System Programmer Management Overview

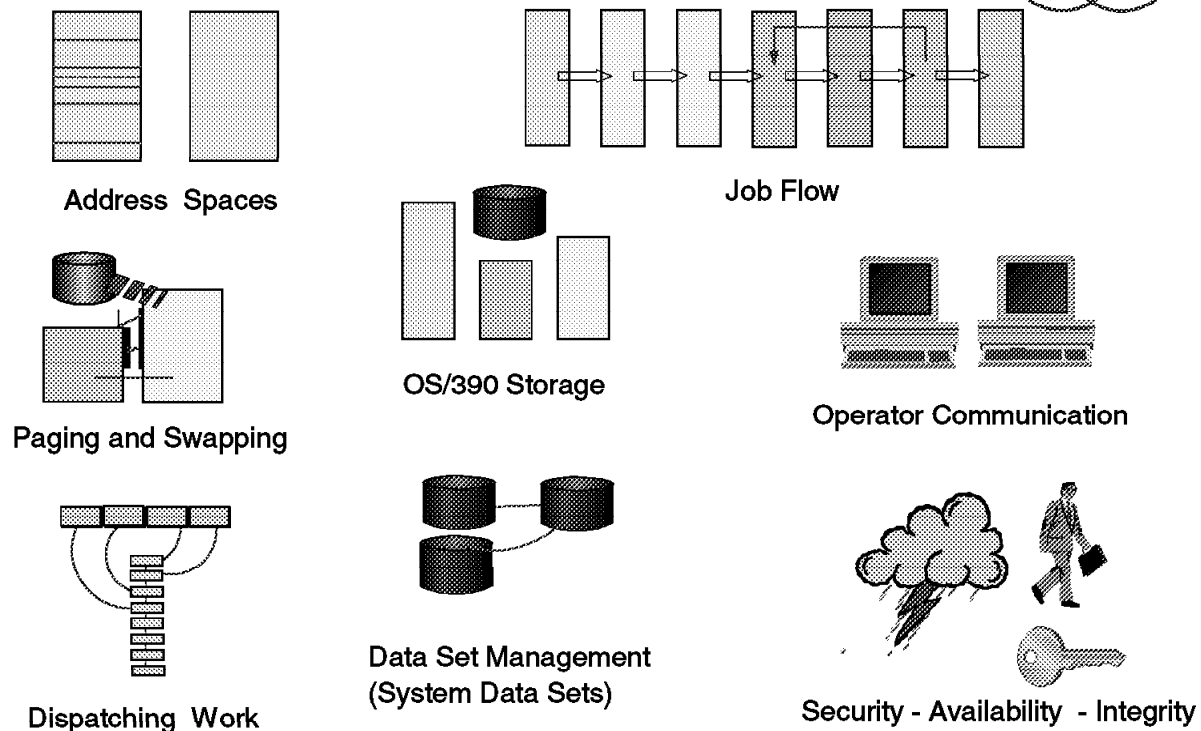


Figure 19. OS/390 System Programmer Management Overview

1.5 OS/390 system programmer functions

As an OS/390 system programmer, you need to be involved in the customization of the items shown on the visual. These items are as follows:

- Address Spaces** When you start the OS/390 operating system, OS/390 establishes system component address spaces. The most important address space is the master scheduler (MSTR). There are other system address spaces for various subsystems and system components.
- Paging and Swapping** Page or swap data sets contain the paged-out portions of address spaces, the common service area (CSA), and the data written to virtual I/O (VIO) data sets.

To define the page and swap data sets, use the access method services DEFINE command.
- Dispatching Work** The scheduling of address spaces and other tasks to execute in the OS/390 system is done by the MVS dispatcher. The MVS dispatcher performs two major system functions. It is responsible for finding and dispatching the highest priority unit of work in the system (SRB, Task, or Interrupted Local Supervisor Routine) and saving status for locked and unlocked tasks and SRBs.

| | |
|-------------------------------|---|
| Job Flow | MVS uses a job entry subsystem (JES) to receive jobs into the operating system, to schedule them for processing by MVS, and to control their output processing. JES is the component of the operating system that provides supplementary job management, data management, and task management functions such as scheduling, control of job flow, and spooling. |
| OS/390 Storage | <p>The system programmer must be aware of all storage considerations when installing and customizing an OS/390 operating system environment. The initialization process begins when the system operator selects the LOAD function at the system console. MVS locates all of the usable central storage that is online and available to the system, and creates a virtual environment for the building of various system areas.</p> <p>This initialization phase allocates the system's minimum virtual storage for the system queue area (SQA) and the extended SQA, allocates virtual storage for the extended local system queue area (extended LSQA) for the master scheduler address space, and allocates virtual storage for the common service area (CSA) and the extended CSA. The amount of storage allocated depends on the values specified on the CSA system parameter at IPL.</p> |
| System Data Sets | Each installation must incorporate required system data sets into the system by allocating space for them on appropriate direct access devices during system installation. The DEFINE function of access method service is used to define both the storage requirements and the volume for each system data set. Some data sets must be allocated on the system residence volume, while some can be placed on other direct access volumes. |
| Operator Communication | <p>The operation of an MVS system involves the following:</p> <ul style="list-style-type: none"> • Console operations, or how operators and system programmers interact with MVS to monitor or control the hardware and software • Message and command processing that forms the basis of operator interaction with MVS and the basis of MVS automation <p>Operating MVS involves managing hardware such as processors and peripheral devices (including the consoles where operators or system programmers do their work) and software such as the MVS operating control system, the job entry subsystem, subsystems such as NetView that can control automated operations and all the applications that run on MVS.</p> |
| Security | Data security is the protection of data against unauthorized disclosure, transfer, modification, or destruction, whether accidental or intentional. A security system must be installed in your operating system by a system programmer to maintain the resources necessary to meet the security objectives. It is the system programmer who has the overall responsibility, using the technology available, to transform the objectives of the security policy into a usable plan. |
| Availability | The software products supporting system programmers and operators in managing their systems heavily influence the complexity of their job and their ability to keep system availability at a high level. Performance management is the system management discipline that most directly impact all users of system resources in an enterprise. You can do this with RMF. |
| Integrity | An operating system is said to have system integrity when it is designed, implemented and maintained to protect itself against unauthorized access, and does so to the extent that security controls specified for that system cannot be compromised. Specifically for MVS, this means that there must be no way for any unauthorized program, using any system interface, defined or undefined: |

- To bypass store or fetch protection
- To bypass OS Password, VSAM Password, or RACF security checking
- To obtain control in an authorized state

System Programmer and Operations

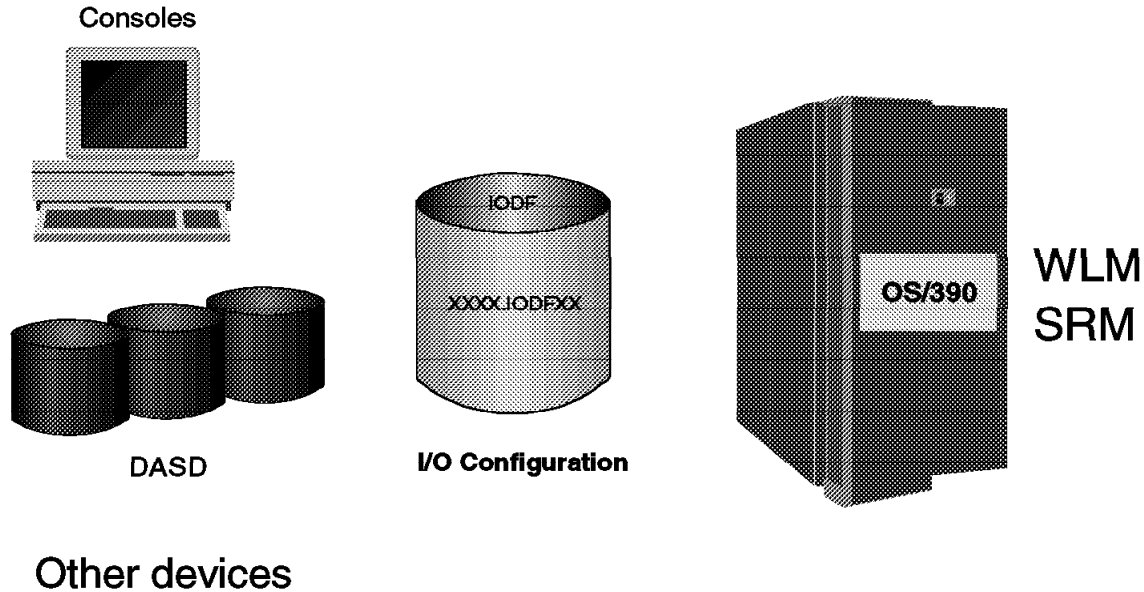


Figure 20. System Programmer and Operations

1.5.1 System programmer and OS/390 operations

A system programmer has to plan the following operations areas:

- **Workload Manager**

MVS Workload Manager provides a solution for managing workload distribution, workload balancing, and distributing resources to competing workloads. MVS Workload Manager is the combined cooperation of various subsystems (CICS, IMS/ESA, JES, APPC, TSO/E, OS/390 UNIX System Services, DDF, DB2, SOM, LSFM, and Internet Connection Server) with the MVS Workload Manager (WLM) component.

- **System performance**

The task of tuning a system is an iterative and continuous process. The controls offered by SRM are only one aspect of this process. Initial tuning consists of selecting appropriate parameters for various system components and subsystems. Once the system is operational and criteria have been established for the selection of jobs for execution via job classes and priorities, SRM will control the distribution of available resources according to the parameters specified by the installation.

SRM, however, can only deal with available resources. If these are inadequate to meet the needs of the installation, even optimal distribution may not be the answer -- other areas of the system should be examined to determine the possibility of increasing available resources.

When requirements for the system increase and it becomes necessary to shift priorities or acquire additional resources, such as a larger processor, more storage, or more terminals, the SRM parameters might have to be adjusted to reflect changed conditions.

- **I/O device management**

You must define an I/O configuration to the operating system (software) and the channel subsystem (hardware). The Hardware Configuration Definition (HCD) component of MVS consolidates the hardware and software I/O configuration processes under a single interactive end-user interface. The validation checking that HCD does as you enter data helps to eliminate errors before you attempt to use the I/O configuration. The output of HCD is an I/O definition file (IODF), which contains I/O configuration data. An IODF is used to define multiple hardware and software configurations to the MVS operating system. When you activate an IODF, HCD defines the I/O configuration to the channel subsystem and/or the operating system. With the HCD activate function or the MVS ACTIVATE operator command, you can make changes to the current configuration without having to initial program load (IPL) the software or power-on reset (POR) the hardware. Making changes while the system is running is known as dynamic configuration or dynamic reconfiguration.

- **Console operations**

The operation of an MVS system involves the following:

- Console operations or how operators interact with MVS to monitor or control the hardware and software.
- Message and command processing that forms the basis of operator interaction with MVS and the basis of MVS automation.

Operating MVS involves managing hardware such as processors and peripheral devices (including the consoles where your operators do their work) and software such as the MVS operating control system, the job entry subsystem, subsystems such as NetView that can control automated operations and all the applications that run on MVS.

Planning MVS operations for a system must take into account how operators use consoles to do their work and how you want to manage messages and commands. Because messages are also the basis of automated operations, understanding message processing in an MVS system can help you plan MVS automation.

1.5.1.1 Managing Operations

Also involved are the business goals and policies established by the installation to allow the installation to grow and handle work efficiently. These needs, of course, vary from installation to installation, but they are important when you plan your MVS operations.

Managing the complexity of MVS requires you to think about the particular needs of the installation. However, any installation might consider the following goals when planning its MVS operations:

- **Increasing system availability**

Many installations need to ensure that their system and its services are available and operating to meet service level agreements. Installations with 24-hour, 7-day operations need to plan for minimal disruption of their operation activities. In terms of MVS operations, how the installation establishes console recovery or whether an operator must re-IPL a system to change processing options are important planning considerations.

- **Controlling operating activities and functions**

As more installations make use of multisystem environments, the need to coordinate the operating activities of those systems becomes crucial. Even for single MVS systems, an installation needs to think about controlling communication between functional areas (such as a tape-pool library and the master console area, for example). In both single and multisystem environments, the

commands operators can issue from consoles can be a security concern that requires careful coordination. As a planner, you want to make sure that the right people are doing the right tasks when they interact with MVS. If your installation uses remote operations to control target systems, you also need to decide about controlling those activities from the host system.

- **Simplifying operator tasks**

Because the complexity of operating MVS has increased, an installation needs to think about the tasks and skills of its operators. How operators respond to messages at their consoles and how you can reduce or simplify their actions are important to operations planning. Also, your installation needs to plan MVS operator tasks in relation to any automated operations that help simplify those tasks.

- **Streamlining message flow and command processing**

In thinking about operator tasks, an installation needs to consider how to manage messages and commands. Operators need to respond to messages. Routing messages to operator consoles, suppressing messages to help your operators manage increased message traffic, or selecting messages for automated operations can all help you manage system activity efficiently.

- **Single system image**

Single system image allows the operator, for certain tasks, to interact with several images of a product as though they were one image. For example, the operator can issue a single command to all MVS systems in the sysplex instead of repeating the command for each system.

- **Single point of control**

Single point of control allows the operator to interact with a suite of products from a single workstation. An operator can accomplish a set of tasks from a single workstation, thereby reducing the number of consoles the operator has to manage.

Requirements for Install



- ★ TSO/E and ISPF
 - ▶ Batch job JCL
- ★ Storage concepts
- ★ Device I/O configurations
- ★ Processor configurations
- ★ Console definitions
- ★ System libraries management
- ★ DASD space management
- ★ Customization parameters - SYS1.PARMLIB
- ★ Data set placement

Figure 21. Requirements for Install

1.5.2 Requirements for install

To be able to install and customize an OS/390 operating system, a system programmer has to know certain basic skills and functions. Using these skills is documented in this book. The visual lists the following areas about which a programmer needs to know:

TSO/E TSO/E is Time Sharing Option Extensions. It is an option of the OS/390 operating system that allows users to interactively share computer time and resources.

TSO/E is a base interactive interface that provides system programmers, non-DP professionals, end users, application programmers, and administrators with an extensive set of commands, services, facilities and programming languages to do productive work on OS/390, and helps to ease systems management. TSO/E is an integral part of OS/390, and serves as a platform for other elements, such as BookManager READ/MVS, HCD, and ISPF.

ISPF/PDF The Interactive System Productivity Facility (ISPF) and its Program Development Facility (ISPF/PDF) work together with TSO/E to provide panels with which users can interact. ISPF provides the underlying dialog management service that displays panels and enables a user to navigate through the panels. ISPF/PDF is a dialog of ISPF that helps maintain libraries of information in TSO/E and allows a user to manage the library through facilities such as browse, edit, and utilities.

Batch JCL During the install phase of OS/390, many batch jobs are required to be submitted. The JCL for these jobs need to be updated for your environment. Therefore, it is essential that a system programmer be very familiar with JCL and batch job submission from TSO/E and using ISPF.

Storage Storage concepts must be understood by the system programmer in setting up an OS/390 environment.

Device I/O An I/O configuration is the hardware resources available to the operating system and the connections between these resources. The resources include:

- Channels
- ESCON Directors (switches)
- Control units
- Devices

When you define a configuration, you need to provide both physical and logical information about these resources. For example, when defining a device you provide physical information, such as its type and model, as well as logical information such as the identifier you will assign in the configuration definition.

You must define an I/O configuration to the operating system (software) and the channel subsystem (hardware). The Hardware Configuration Definition (HCD) component of MVS consolidates the hardware and software I/O configuration processes under a single interactive end-user interface.

Processors When more than one processor exists in a complex or more than one logical partition exists in a complex, OS/390 is required to be defined in multisystem mode or a sysplex.

Consoles A console configuration consists of the various consoles that operators use to communicate with OS/390. Your installation first defines the I/O devices it can use as consoles with the hardware configuration definition (HCD). HCD manages the I/O configuration for the OS/390 system. Once you have defined the devices, indicate to OS/390 which devices to use as consoles by specifying the appropriate device numbers in the CONSOLxx parmlib member.

Installing OS/390



- ★ IBM and other installation packages
- ★ Maintenance of OS/390 using SMP/E
- ★ Choosing an install package

Figure 22. Installing OS/390

1.6 Installing OS/390

Because the base elements and optional features of OS/390 are integrated into a single package with compatible service levels, you must install, with few exceptions, the entire OS/390 product.

You can install OS/390 using one of several IBM packages. Two of these packages are available at no additional charge when you license OS/390:

- ServerPac
- CBPDO

Other installation packages and offerings are available for a fee.

When you order a new system or a new release of OS/390, you also receive all the new maintenance or service that is applicable to the release.

1.6.1 Maintenance or service using SMP/E

All executable operating system code is subject to errors. Any errors in the code are fixed by IBM and made available to installations in the form of either an authorized program analysis report (APAR) or program temporary fix (PTF). IBM provides a product to help you maintain or apply maintenance to your system. It is called System Modification Program Extended (SMP/E).

SMP/E is a tool designed to manage the installation of software products on your OS/390 system, and

to track the modifications you make to those products. Usually, it is the system programmer's responsibility to ensure that all software products and their modifications are properly installed on the system. The system programmer also has to ensure that all products are installed at the proper level so all elements of the system can work together. At first, that might not sound too difficult, but as the complexity of the software configuration increases, so does the task of monitoring all the elements of the system.

1.6.2 Choosing an install package

The best installation method is usually the one that requires the least amount of work for you. IBM recommends the following:

- If you are new to OS/390 and never had a previous system, use either a fee service or ServerPac's full system replacement option.
- If you're migrating from VM or VSE, use a fee service.
- If you're migrating from a level of products available before the general availability of MVS/ESA SP 4.3 (June of 1992), or if you're running unsupported levels of products, use a fee service or ServerPac's full system replacement.
- If you're migrating from MVS/ESA SP 4.3 or above, or any release of OS/390, use any method (ServerPac, CBPDO, or fee service).

Installing OS/390 - ServerPac

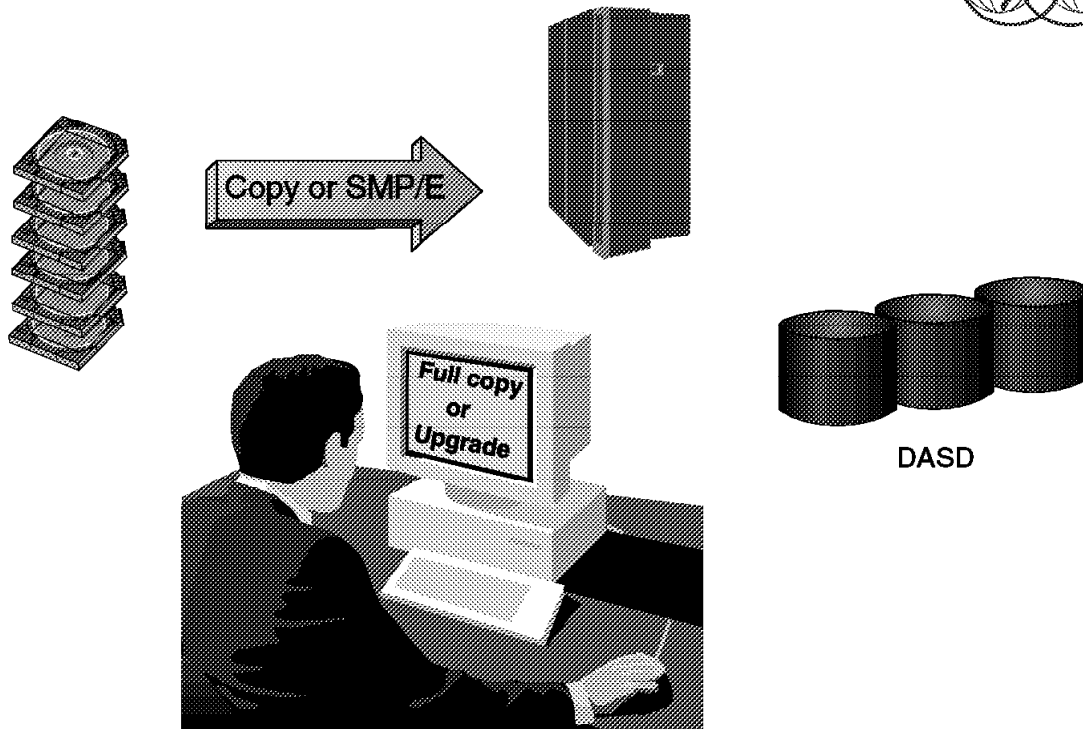


Figure 23. Installing OS/390 - ServerPac

1.6.3 Installing OS/390 using ServerPac

ServerPac is a software delivery package consisting of products and service for which IBM has performed the SMP/E installation steps and some of the post-SMP/E installation steps. To install the package on your system and complete the installation of the software it includes, you use the CustomPac Installation Dialog.

An OS/390 ServerPac order contains an Interactive System Productivity Facility (ISPF) dialog that you use to install OS/390. This dialog is called the CustomPac Installation Dialog because it is used to install all of IBM's CustomPac offerings, for example, ServerPac, SystemPac, FunctionPac, ProductPac, and ServicePac.

When ordering a ServerPac:

- IBM selects the products that were ordered.
- IBM integrates products and selected service into target and distribution libraries and their SMP/E zones.
- IBM enables the features that you ordered that use dynamic enablement.
- IBM verifies the resulting system for the specific package by doing an IPL, submitting a job, logging on to TSO/E, and checking the job's output. IBM performs this test using the operational data sets supplied with the ServerPac. If you use the software upgrade path when installing a ServerPac, you will use your own existing operational data sets, so this test will not assure that the system

will IPL in your environment. However, it does make sure that the software itself can be used to IPL given a usable set of operational data sets.

- IBM selects all unintegrated service for products ordered and includes it on a service tape.

Installing OS/390 - CBPDO Custom-Built Product Delivery Option

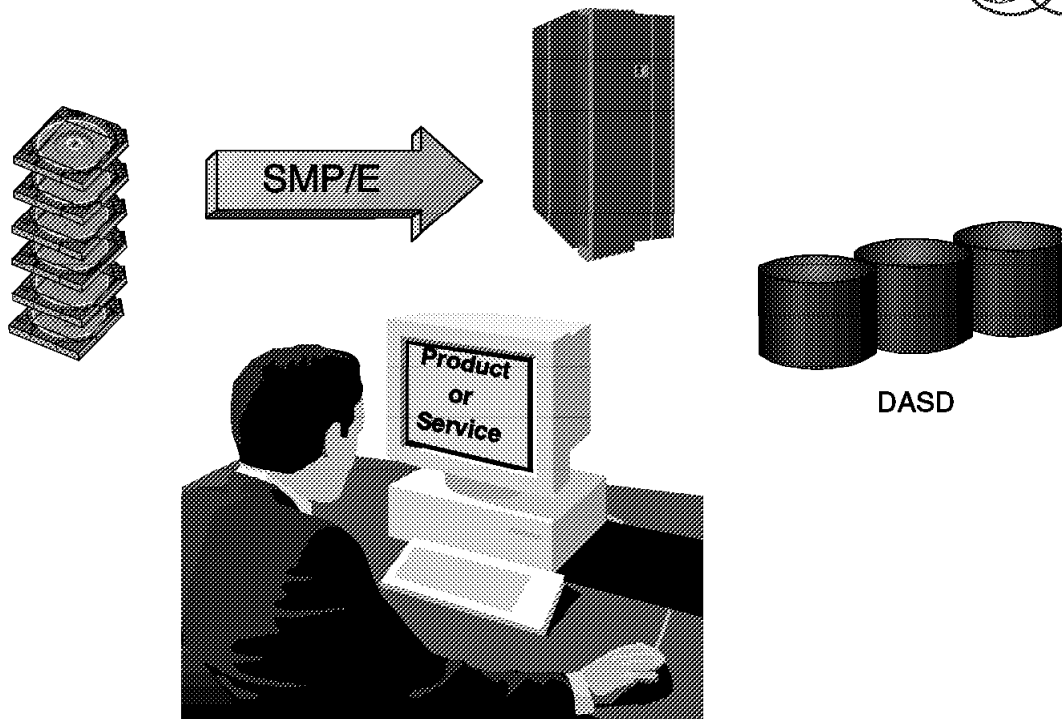


Figure 24. Installing OS/390 - CBPDO

1.6.4 Installing OS/390 using CBPDO

The Custom-Built Product Delivery Option (CBPDO) is a software delivery package consisting of uninstalled products and unintegrated service. You must use SMP/E to install the individual OS/390 elements and features, and their service, before you can IPL.

To order CBPDO, use an order checklist (available from an IBM representative, the OS/390 Web site, or IBMLink via the configurator). The order is for a unique system release identifier (SREL). IBM recommends that you order all products that you maintain in the same OS/390 product set.

To order CBPDO, use an order checklist (available from an IBM representative, the OS/390 Web site, or IBMLink via the configurator). The order is for a unique system release identifier (SREL). IBM recommends that you order only OS/390 elements and features (or their equivalent stand-alone products) in your CBPDO order. (Ordering equivalent levels of nonexclusive elements does not increase the size of the CBPDO because the FMIDs are the same, but it does enable the IFAPRD00 PARMLIB member to be built correctly.) If you need to update other products, place a separate CBPDO order for these products.

When ordering a ServerPac:

- IBM selects the products that were ordered.
- IBM selects service for the products you order and for products that are already licensed under the same customer number you use to place your order.

- IBM builds a customized job stream to enable the features that you ordered that use dynamic enablement.
- IBM builds a customized job stream to enable selected features that use the registration service.

Installing OS/390 - Fee-Based Packages

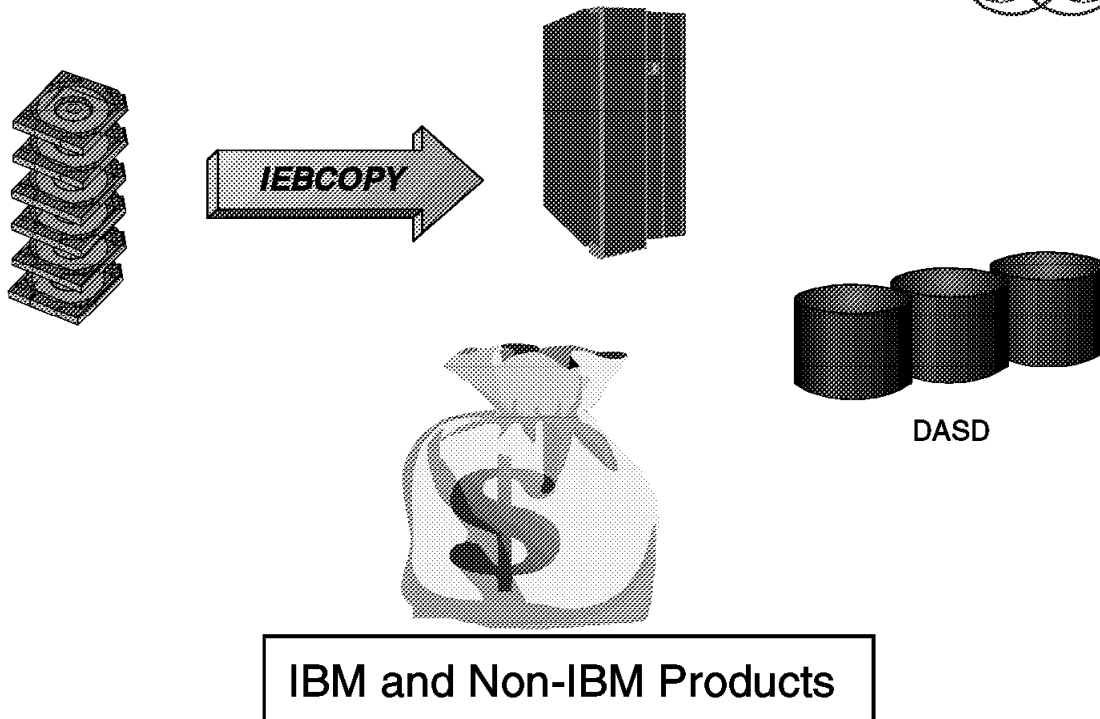


Figure 25. Installing OS/390 Using Fee-Based Packages

1.6.5 Installing OS/390 using fee-based packages

Instead of using ServerPac or CBPDO to install OS/390, you could, for an additional fee, use one of the following services:

- SystemPac tailors OS/390 to your environment (such as DASD layout, migration of MVSCP/IOCP to IODF, and naming conventions) based on information provided to IBM. With this offering, selected non-IBM products can be integrated. This offering can be delivered in IEBCOPY dump-by-data-set format or in full volume dump format.
- SoftwareXcel Installation Express (SIE), available in the U.S. only, provides prebuilt OS/390 system packages in full volume dump format, tailored to customer hardware and software configurations. SIE includes on-site planning, installation, and package testing. SIE creates a compatibility research report for up to 70 non-IBM software products if requested. SIE can also include selected non-IBM software products integrated into the system package.
- The Entry Server Offering (available in some countries) is a packaged solution that includes hardware, software, installation services, maintenance and financing to help customers get to current technology.
- Other fee-based help includes Washington System Center services, customized solutions, hardware services, and software services.

Chapter 2. OS/390 Storage Concepts

This chapter describes many of the OS/390 storage concepts that system programmers need to know to do their job. Many of the concepts needed by system programmers to do their job are as follows:

- Address spaces
- Subsystem definitions
- Virtual storage layouts for address spaces
- How storage is managed by OS/390
- How processor storage is managed

The initialization process begins when the system operator selects the LOAD function at the system console. MVS locates all of the usable central storage that is online and available to the system, and creates a virtual environment for the building of various system areas.

Processor storage consists of central storage plus expanded storage. The system uses a portion of both central storage and virtual storage. To determine how much central storage is available to the installation, the system's fixed storage requirements must be subtracted from the total central storage. The central storage available to an installation can be used for the concurrent execution of the paged-in portions of any installation programs.

To tailor the system's storage parameters, you need a general understanding of the system initialization and storage initialization processes.

The system initialization process prepares the system control program and its environment to do work for the installation. The process essentially consists of:

- System and storage initialization, including the creation of system component address spaces.
- Master scheduler initialization and subsystem initialization.

When the system is initialized and the job entry subsystem is active, the installation can submit jobs for processing by using the START, LOGON, or MOUNT command.

In addition to initializing system areas, MVS establishes system component address spaces. MVS establishes an address space for the master scheduler (the master scheduler address space) and other system address spaces for various subsystems and system components. Some of the component address spaces are:

- Program call/authorization for cross-memory communications
- System trace
- Global resource serialization
- Dumping services.

OS/390 Address Spaces

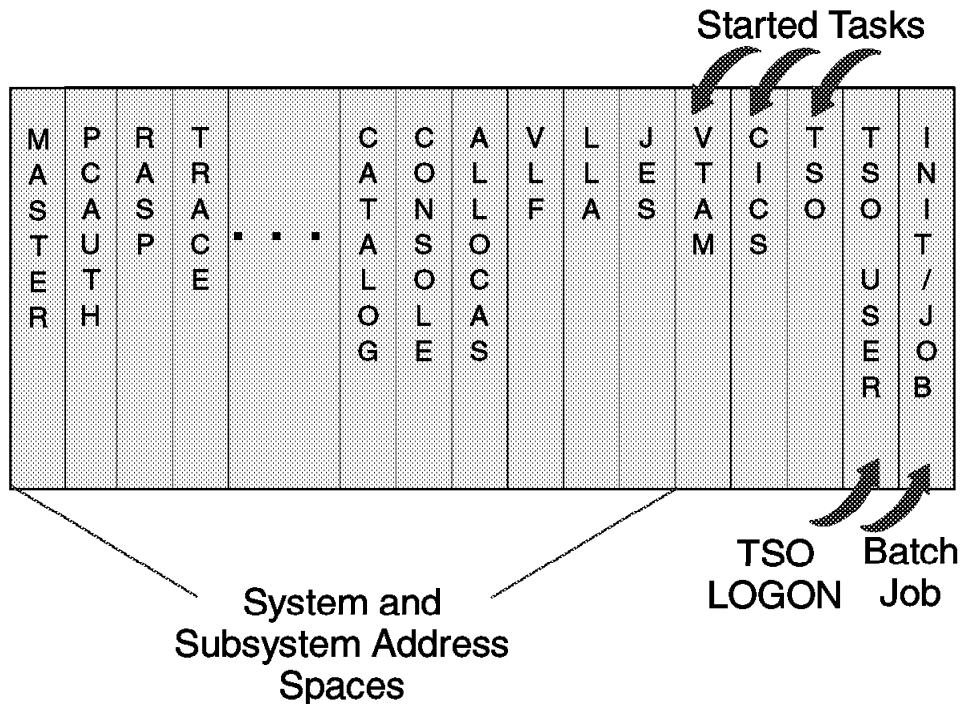


Figure 26. OS/390 address spaces

2.1 OS/390 Address Spaces

When you start OS/390, master scheduler initialization routines initialize system services such as the system log and communications task, and start the master scheduler address space, which becomes address space number one (ASID=1). Other system address spaces are then started during the initialization process of OS/390.

Then the subsystem address spaces are started. The master scheduler starts the job entry subsystem (JES2 or JES3). JES is the primary job entry subsystem. Then other defined subsystems are started. All subsystems are defined in SYS1.PARMLIB, member IEFSSNxx. These subsystems are secondary subsystems.

The visual shows four types of address spaces as follows:

- System** The system address spaces are started following initialization of the master scheduler. These address spaces perform functions for all the other types of address spaces that start in an OS/390 system.
- Subsystem** You cannot run OS/390 without a primary job entry subsystem, either JES2 or JES3.
- TSO logon** These address spaces start when a user issues a logon to TSO/E. Each user executes in a separate address space.
- Batch job** These address spaces are started by JES when a JCL stream is passed to JES and a job is created and then subsequently scheduled into execution.

Subsystem Definitions



SYS1.PARMLIB: IEFSSNxx

```
SUBSYS      SUBNAME (subname)
             [CONSNAME (consname) ]
             [INITRTN (initrtn)
             [INITPARM (initparm) ] ]
             [PRIMARY ( {NO | YES} )
             [START ( {YES | NO} ) ] ]
```

```
SUBSYS SUBNAME(JES2) PRIMARY(YES)
```

```
SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN)
        INITPARM('ID=60,PROMPT=YES')
```

SYS1.PARMLIB: IEASYSxx SSN=xx

Figure 27. Subsystem definitions

2.1.1 Subsystem definitions

Subsystem initialization is the process of readying a subsystem for use in the system. IEFSSNxx members of SYS1.PARMLIB contain the definitions for the primary subsystems, such as JES2 or JES3, and the secondary subsystems, such as SMS and DB2. For detailed information about the data contained in IEFSSNxx members for secondary systems, refer to the installation manual for the specific subsystem. IEFSSNxx allows you to specify the following:

- The subsystem initialization routine to be given control during master scheduler initialization.
- The input parameter string to be passed to the subsystem initialization routine.
- A primary subsystem name and whether you want it started automatically.

The order in which the subsystems are initialized depends on the order in which they are defined in the IEFSSNxx parmlib member on the SSN parameter. Unless you are starting the storage management subsystem (SMS), start the primary subsystem (JES) first.

Note: The storage management subsystem (SMS) is the only subsystem that can be defined before the primary subsystem.

Some subsystems require the services of the primary subsystem in their initialization routines. Problems can occur if subsystems that use the subsystem affinity service in their initialization routines are initialized before the primary subsystem. If you are starting SMS, specify its record before you specify the primary subsystem record.

Note: In general, it is a good idea to make the subsystem name the same as the name of the member of SYS1.PROCLIB used to start the subsystem. If the name does not match, you may receive error messages when you start the subsystem.

The SSN parameter in IEASYSxx identifies the IEFSSNxx member that the system is to use to initialize the subsystems, as follows:

```
SSN=      {aa          }  
          {(aa,bb,...) }
```

The two-character identifier, represented by aa (or bb, and so forth), is appended to IEFSSN to identify IEFSSNxx members of parmlib. If the SSN parameter is not specified, the system uses the IEFSSN00 parmlib member.

The order in which the subsystems are defined on the SSN parameter is the order in which they are initialized. For example, a specification of SSN=(13,Z5) would cause those subsystems defined in the IEFSSN13 parmlib member to be initialized first, followed by those subsystems defined in the IEFSSNZ5 parmlib member.

Note: If you specify duplicate subsystem names in IEFSSNxx parmlib members, the system issues message IEFJ003I to the SYSLOG, the master console, and consoles that monitor routing code 10 messages.

S/390 Storage

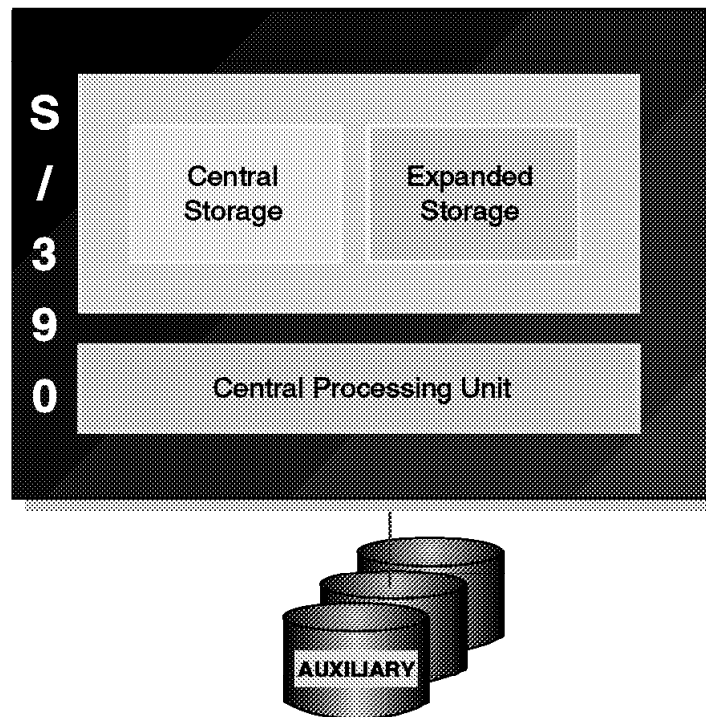


Figure 28. S/390 storage

2.2 Processor Storage Overview

Processor storage consists of central storage plus expanded storage. The system uses a portion of both central storage and virtual storage. To determine how much central storage is available to the installation, the system's fixed storage requirements must be subtracted from the total central storage. The central storage available to an installation can be used for the concurrent execution of the paged-in portions of any installation programs.

Note: Each installation is responsible for establishing many of the central storage parameters that govern RSM's processing.

Central Central storage often referred to as main storage, provides the system with directly addressable fast-access storage of data. Both data and programs must be loaded into central storage (from input devices) before they can be processed. Main storage may include one or more smaller faster-access buffer storages, sometimes called caches. A cache is usually physically associated with a CPU or an I/O processor. The effects, except on performance, of the physical construction and use of distinct storage media are not observable by the program.

Expanded Expanded storage may be available on some models. Expanded storage, when available, can be accessed by all CPUs in the configuration by means of instructions that transfer 4 KB blocks of data from expanded storage to main storage or from main storage to expanded storage. Each 4 KB block in expanded storage is addressed by means of a 32-bit unsigned binary integer called an expanded-storage block number.

CPU

The central processing unit (CPU) is the controlling center of the system. It contains the sequencing and processing facilities for instruction execution, interruption action, timing functions, initial program loading and other machine-related functions.

The physical implementation of the CPU may differ among models, but the logical function remains the same. The result of executing an instruction is the same for each model, providing that the program complies with the compatibility rules.

The CPU, in executing instructions, can process binary integers and floating-point numbers of fixed length, decimal integers of variable length, and logical information of either fixed or variable length. Processing may be in parallel or in series; the width of the processing elements, the multiplicity of the shifting paths, and the degree of simultaneity in performing the different types of arithmetic differ from one CPU to another without affecting the logical results.

Auxiliary

An installation needs auxiliary direct access storage devices (DASD) for placement of all system data sets. Enough auxiliary storage must be available for the programs and data that comprise the system. Auxiliary storage used to support basic system requirements has three logical areas as follows:

- System data set storage area
- Paging data sets for backup of all pageable address spaces
- Swap data sets used for LSQA pages and private area pages that are swapped in with the address space (also called the *working set*)

Storage Managers

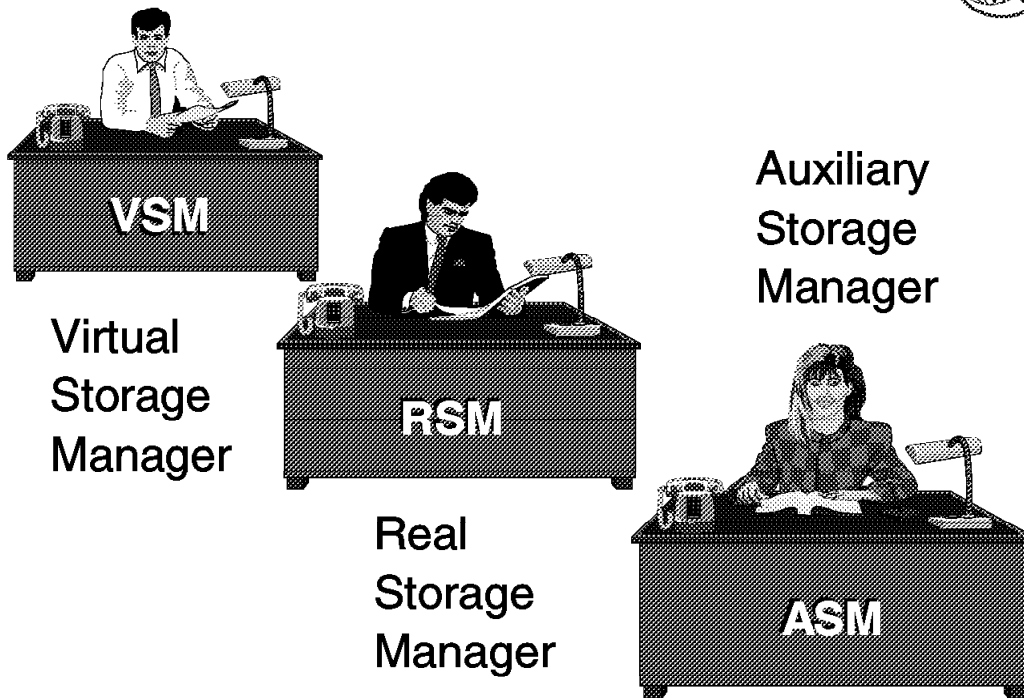


Figure 29. Storage managers

2.3 Storage managers

In an OS/390 system, storage is managed by the following storage components managers:

- Real** The real storage manager (RSM) controls the allocation of central storage during initialization and pages in user or system functions for execution. Some RSM functions:
- Allocate central storage to satisfy GETMAIN requests for SQA and LSQA
 - Allocate central storage for page fixing
 - Allocate central storage for an address space that is to be swapped in
 - Allocate and initialize control blocks and queues related to expanded storage
- Virtual** Each installation can use virtual storage parameters to specify how certain virtual storage areas are to be allocated. These parameters have an impact on central storage use and overall system performance.
- Auxiliary** The auxiliary storage manager code controls the use of page and swap data sets. As a system programmer, you are responsible for:
- Page and swap operations
 - Page and swap data set sizes
 - Space calculation
 - Performance of page and swap data sets
 - Estimating the total size of the paging data sets

Virtual Storage Manager

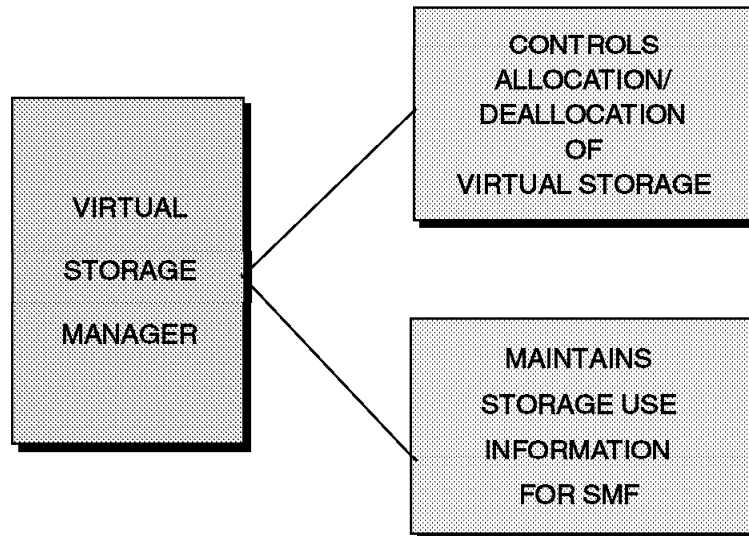


Figure 30. Virtual storage manager

2.3.1 Virtual Storage Manager

Virtual storage is managed by the virtual storage manager (VSM). Its main function is to distribute the virtual storage among all requests.

Virtual storage is requested with the GETMAIN or STORAGE OBTAIN macro and returned to the virtual storage manager with the FREEMAIN or STORAGE RELEASE macro.

Virtual Storage

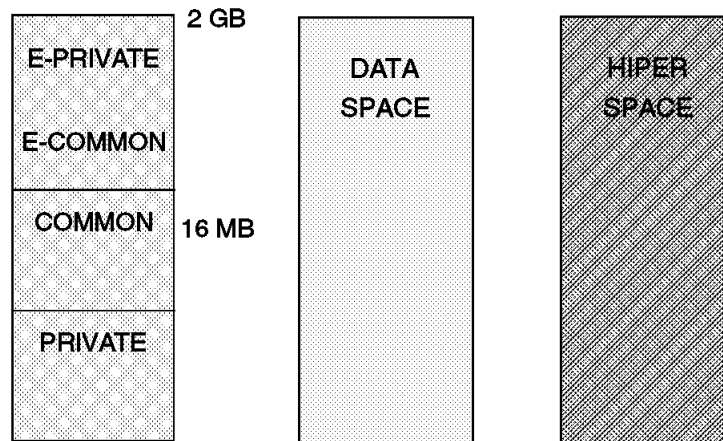


Figure 31. Virtual storage

2.3.2 Virtual Storage

Virtual storage is normally larger than main storage (called real storage in OS/390). The size of real storage depends on the CPU type. In a computing system *without* virtual storage, a program *cannot* be executed unless there is enough storage to hold it. In addition the complete storage used is allocated until it is finished.

An OS/390 program resides in virtual storage and only parts of the program currently active need to be in real storage at processing time. The inactive parts are held in auxiliary storage, DASD devices, called page data sets. An active virtual storage page resides in a real storage frame. An inactive virtual storage page resides in a auxiliary storage slot. Moving pages between frames and slots is called *paging*.

2.3.2.1 Estimating Virtual Storage

Estimating the virtual storage allocated at an installation is important primarily because this storage must be backed up by central storage in some ratio (for example, 25%). This backup storage contributes significantly to an installation's total central storage requirements.

Virtual storage must also be backed up by expanded storage or auxiliary storage. Each installation can use virtual storage parameters to specify how certain virtual storage areas are to be allocated. These parameters have an impact on central storage use and overall system performance.

2.3.2.2 Virtual Storage Address Space

A two-gigabyte virtual storage address space is provided for:

- The master scheduler address space
- JES
- Other system component address spaces, such as allocation, system trace, system management facilities (SMF), and dumping services
- Each user (batch or TSO/E).

The system uses a portion of each virtual address space. Each virtual address space consists of:

- The common area below 16 megabytes
- The private area below 16 megabytes
- The extended common area above 16 megabytes
- The extended private area above 16 megabytes.

Virtual Storage Address Space

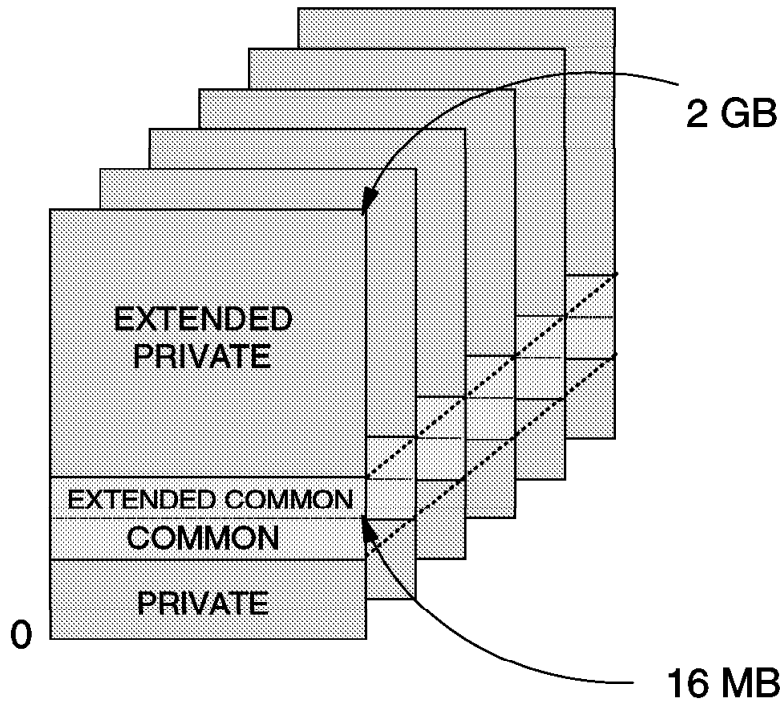


Figure 32. Multiple address spaces

2.3.2.3 Multiple Address Spaces

The system uses a portion of each virtual address space.

Each virtual address space consists of:

- The common area below 16 MB
- The private area below 16 MB
- The extended common area above 16 MB
- The extended private area above 16 MB

Address Space Storage

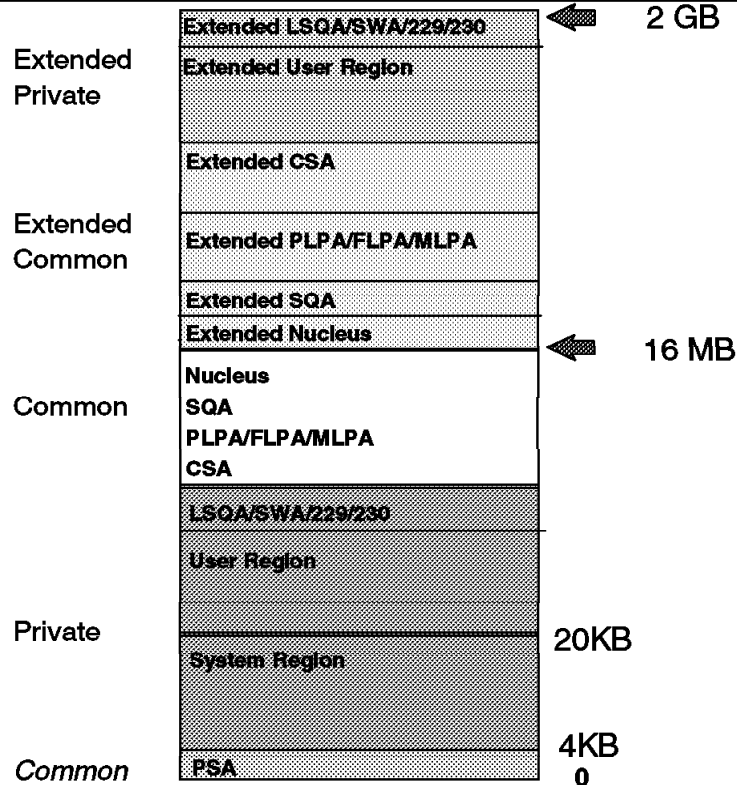


Figure 33. Address space storage

2.4 Address space storage

Estimating the virtual storage allocated at an installation is important primarily because this storage must be backed up by central storage in some ratio (for example, 25%). This backup storage contributes significantly to an installation's total central storage requirements.

Virtual storage must also be backed up by expanded storage or auxiliary storage. Each installation can use virtual storage parameters to specify how certain virtual storage areas are to be allocated. These parameters have an impact on central storage use and overall system performance. The following overview describes the function of each virtual storage area.

A two-gigabyte (2-GB) virtual storage address space is provided for:

- The master scheduler address space
- JES
- Other system component address spaces, such as allocation, system trace, system management facilities (SMF), and dumping services
- Each user (batch or TSO/E).

The system uses a portion of each virtual address space. Each virtual address space consists of:

- The common area below 16 MB
- The private area below 16 MB
- The extended common area above 16 MB
- The extended private area above 16 MB

The common area contains system control programs and control blocks. The following storage areas are located in the common area:

- Prefixed storage area (PSA)
- Common service area (CSA)
- Pageable link pack area (PLPA)
- Fixed link pack area (FLPA)
- Modified link pack area (MLPA)
- System queue area (SQA)
- Nucleus, which is fixed and nonswappable

Each storage area in the common area (below 16 MB) has a counterpart in the extended common area (above 16 MB) with the exception of the PSA.

Each address space uses the same common area. Portions of the common area are paged in and out as the demands of the system change and as new user jobs (batch or time-shared) start and old ones terminate.

The private area contains:

- A local system queue area (LSQA)
- A scheduler work area (SWA)
- Subpools 229, 230, and 249 (the authorized user key area)
- A 16 KB system region area
- Either a V=V (virtual = virtual) or V=R (virtual = real) private user region for running programs and storing data

Except for the 16 KB system region area and V=R user regions, each storage area in the private area below 16 MB has a counterpart in the extended private area above 16 MB.

Each address space has its own unique private area allocation. The private area (except LSQA) is pageable unless a user specifies a V=R region. If assigned as V=R, the actual V=R region area (excluding SWA, the 16 KB system region area, and subpools 229, 230, and 249) is fixed and nonswappable.

Common Versus Private Area

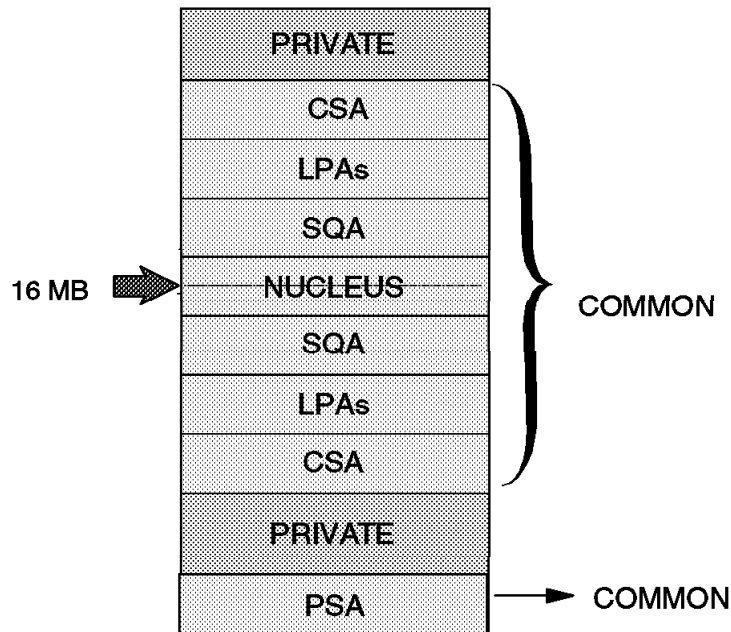


Figure 34. Common versus private area

2.4.1.1 Common Area

The common area contains system control programs and control blocks. The following storage areas are located in the common area:

- Prefixed storage area (PSA)
- Common service area (CSA)
- Pageable link pack area (PLPA)
- Fixed link pack area (FLPA)
- Modified link pack area (MLPA)
- System queue area (SQA)
- Nucleus, which is fixed and nonswappable

Each storage area in the common area (below 16 MB) has a counterpart in the extended common area (above 16 MB) with the exception of the PSA.

Each address space uses the same common area. Portions of the common area are paged in and out as the demands of the system change and as new user jobs (batch or time-shared) start and old ones terminate.

2.4.1.2 Private Area

The private area contains:

- A local system queue area (LSQA).
- A scheduler work area (SWA).
- Subpools 229, 230, and 249 (the authorized user key area).
- A 16 KB system region area.
- Either a V=V (virtual = virtual) or V=R (virtual = real) private user region for running programs and storing data.

Except for the 16 KB system region area and V=R user regions, each storage area in the private area below 16 MB has a counterpart in the extended private area above 16 MB.

Except for the 16 KB system region area and V=R user regions, each storage area in the private area below 16 MB has a counterpart in the extended private area above 16 MB.

System Queue Area (SQA)

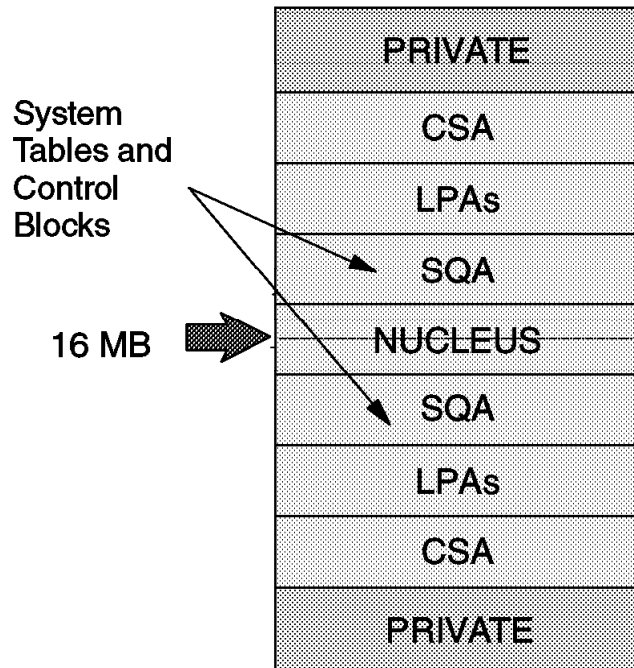


Figure 35. System Queue Area (SQA/Extended SQA)

2.4.1.3 System Queue Area (SQA/Extended SQA)

This area contains tables and queues relating to the entire system. Its contents are highly dependent on configuration and job requirements at an installation. The total amount of virtual storage and number of private virtual storage address spaces are two of the factors that affect the system's use of SQA.

The SQA is allocated directly below the nucleus; the extended SQA is allocated directly above the extended nucleus.

The size of the SQA can be specified through the:

- SQA parameter in the IEASYSxx member of SYS1.PARMLIB
- NIP or operator's console

Virtual SQA is allocated as a number of 64 KB blocks to be added to the minimum system requirements for SQA. If the SQA required by the system configuration exceeds the amount that has been reserved through the SQA parameter, the system attempts to allocate additional virtual SQA from the CSA area. If less than eight frames (below 16 MB) remain for allocation purposes, the system stops creating new address spaces. When SQA is in use, it is fixed in central storage.

The size of the SQA cannot be increased or decreased by the operator during a restart that reuses the previously initialized PLPA (a quick start). The size will be the same as during the preceding IPL.

Common Service Area (CSA)

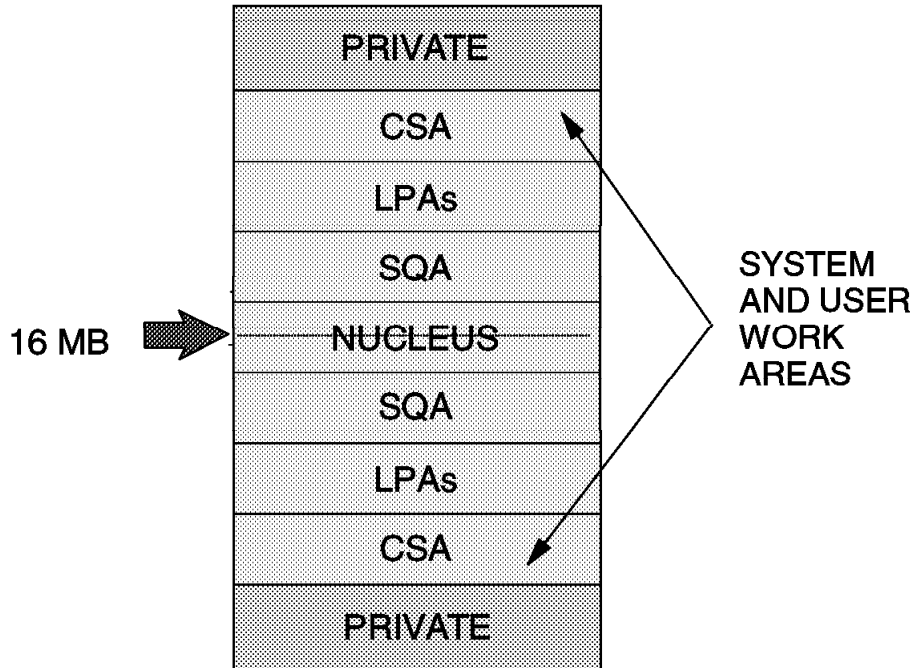


Figure 36. Common service area

2.4.1.4 Common Service Area (CSA/Extended CSA)

This area contains pageable and fixed data areas that are addressable by all active virtual storage address spaces. CSA normally contains data referenced by a number of system address spaces, enabling address spaces to communicate by referencing the same piece of CSA data.

CSA is allocated directly below the MLPA; extended CSA is allocated directly above the extended MLPA. If the virtual SQA space is depleted, the system will allocate additional SQA space from the CSA. The size of the CSA is specified by:

- The SYSP parameter at the operator's console to specify an alternative system parameter list (IEASYSxx) that contains a CSA specification.
- The CSA parameter at the operator's console during system initialization. This value overrides the current system parameter value for CSA that was established by IEASYS00 or IEASYSxx.

Note: If the size allocated for extended SQA is too small or is used up very quickly, the system attempts to steal space from extended CSA. When both extended SQA and extended CSA are used up, the system allocates space from SQA and CSA below 16 MB. The allocation of this storage could eventually lead to a system failure. Ensuring the appropriate size of extended SQA and extended CSA storage is critical to the long-term operation of the system.

Nucleus

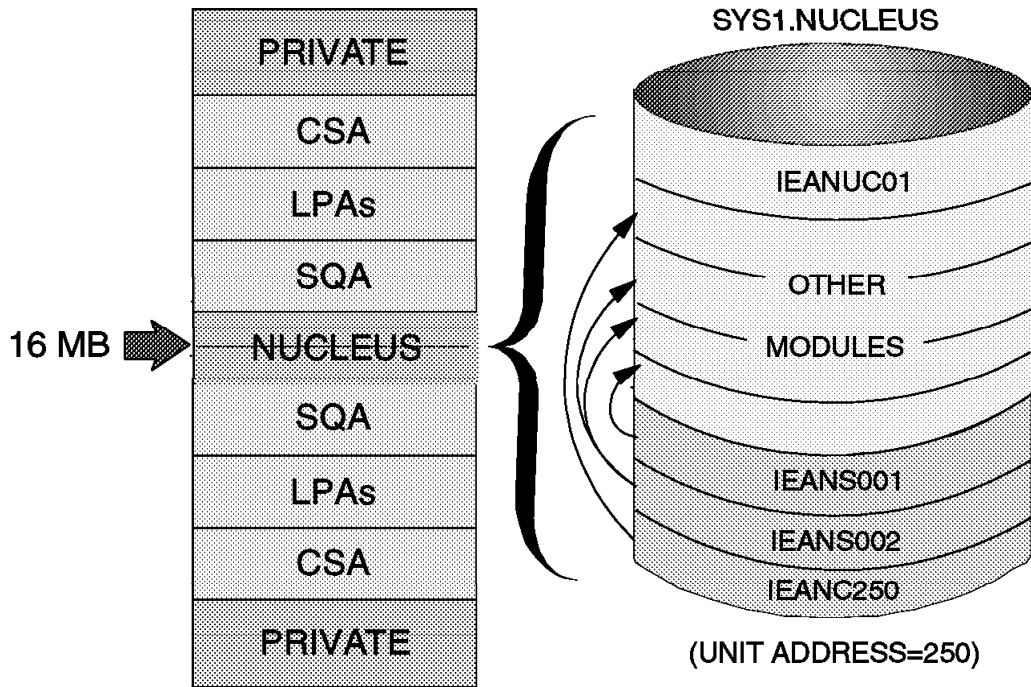


Figure 37. Nucleus

2.4.1.5 Nucleus

The nucleus area contains the nucleus load module and extensions to the nucleus that are initialized during IPL processing.

The modules to be added to the nucleus region, or deleted from it, must reside in members of SYS1.NUCLEUS. To add or delete modules, simply specify the members on INCLUDE or EXCLUDE statements in NUCLSTxx.

The nucleus is always fixed storage.

Region

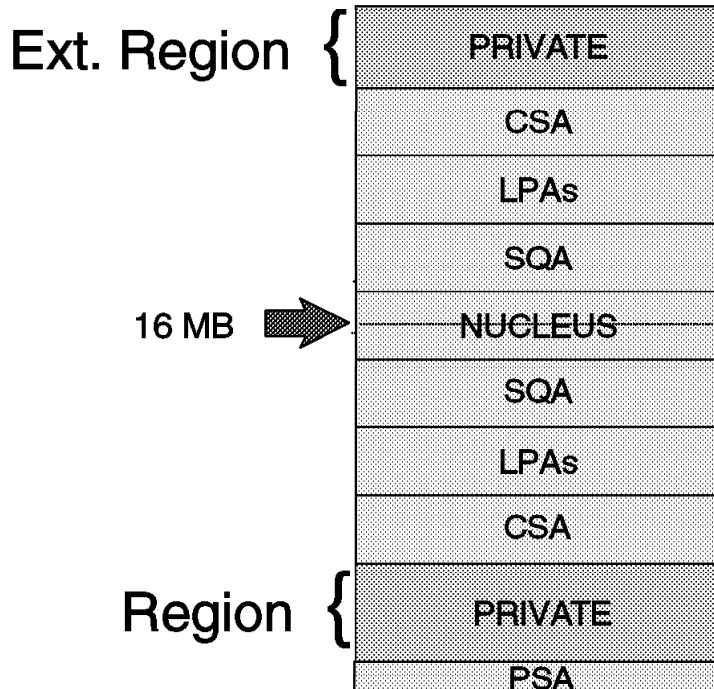


Figure 38. Region

2.4.1.6 Region

The portion of the user's private area within each virtual address space that is available to the user's problem programs is called the user region.

In MVS, the storage available for a program is virtual storage or centra storage (also called real storage):

Virtual storage Virtual storage is addressable space that appears to the user as central (real) storage. Instructions and data are mapped from virtual storage into central storage locations, where they are executed.

Central storage Central (real) storage is the storage from which the processor can directly obtain instructions and data and to which it can directly return results.

The virtual storage address space is 2 gigabytes. The address space contains the commonly addressable system storage, the nucleus, and the private address space, which includes the user's region. When a program is selected, the system brings it into virtual storage and divides it into pages of 4K bytes. The system transfers the pages of a program into central (real) storage for execution and out to auxiliary storage when not needed. Paging is done automatically; to the programmer the entire program appears to occupy contiguous space in central storage at all times. Actually, not all pages of a program are necessarily in central storage at one time. Also, the pages that are in central storage do not necessarily occupy contiguous space.

For a complete explanation about region, refer to *OS/390 Initialization and Tuning Guide*, SC28-1751, and *OS/390 Initialization and Tuning Reference*, SC28-1752.

Data Space

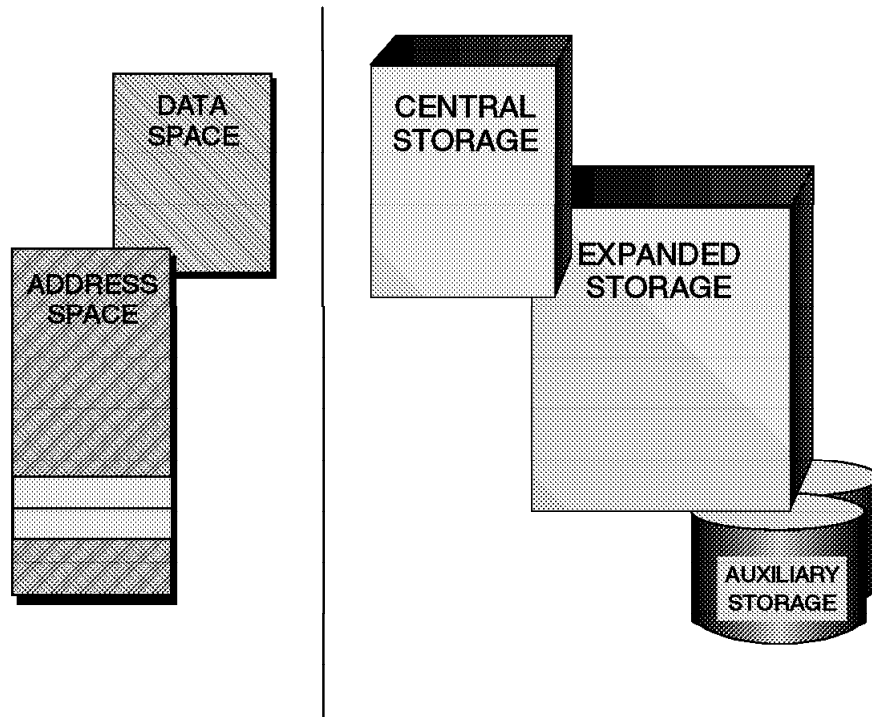


Figure 39. Data space

2.4.1.7 Data Space

Another type of address space is called a *data space*. A data space has a range up to 2 GB of contiguous virtual storage that a user can ask the system for. Within a data space, all addresses, except the first 4 KB (PSA), are available because there are no common areas as in the nucleus, SQA or LSQA. A data space can hold only data; it does not contain MVS control blocks.

Multiprogramming and Multiprocessing

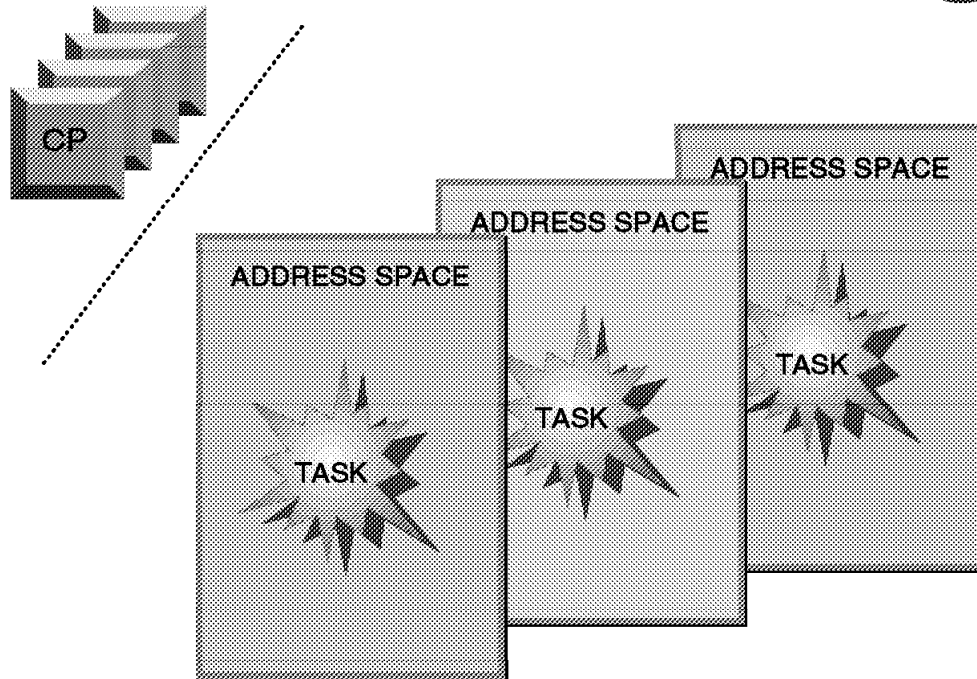


Figure 40. Multiprogramming and multiprocessing

2.4.1.8 Multiprogramming

Many programs may be in the system at the same time, each in its own address space. In a single processor only one of these programs can be active at a time. However, the active program may lose control anytime because of interrupts. The SCP selects which program should get control next.

2.4.1.9 Multiprocessing

Multiprocessing is a logical expansion of multiprogramming. It means the execution of more than one program (task) simultaneously on more than one processor. All processors operate under a single SCP.

Remember:

- Each processor has a current PSW, its own set of registers, and assigned storage locations.
- When a single processor shares real storage with other processors, then all of them are controlled by a single SCP. This is called a *tightly coupled multiprocessing complex*.
- When a single processor shares a common workload with others, but does not share real storage, this is called a *loosely coupled multiprocessing complex*.

Real Storage Manager

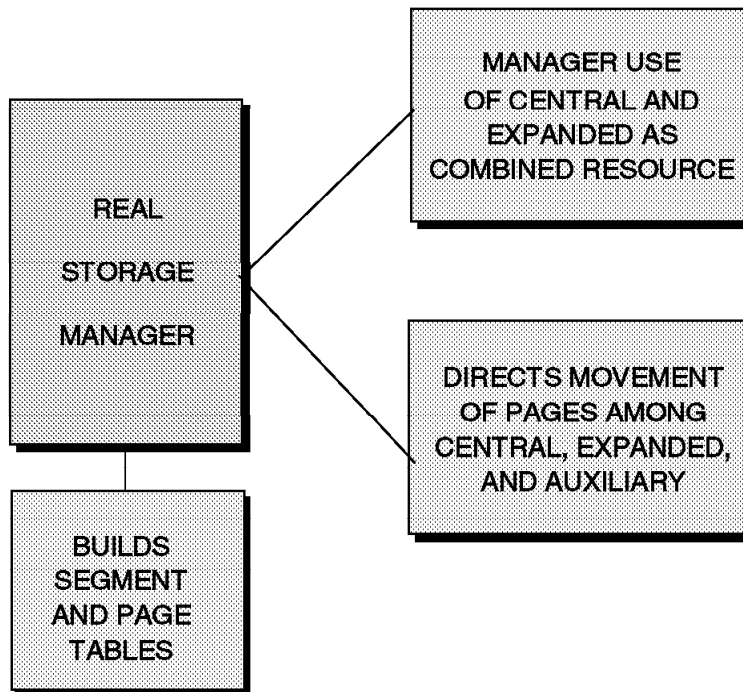


Figure 41. Real storage manager

2.4.2 Real Storage Manager

The task of the Real Storage Manager (RSM) is to control the usage of real storage frames. RSM acts together with the auxiliary storage manager to support the virtual storage concept, and with VSM to ensure that a GETMAINED page will be backed up with a real storage frame. Furthermore, RSM establishes many services to other components and application programs to manipulate the status of pages and frames.

RSM controls the allocation of central storage during initialization and pages in user or system functions for execution. Some RSM functions are to:

- Allocate central storage to satisfy GETMAIN requests for SQA and LSQA
- Allocate central storage for page fixing
- Allocate central storage for an address space that is to be swapped in
- Allocate and initialize control blocks and queues related to expanded storage

If there is storage above 16 megabytes, RSM allocates central storage locations above 16 megabytes for SQA, LSQA, and the pageable requirement of the system. When non-fixed pages are fixed for the first time, RSM:

- Ensures that the pages occupy the appropriate type of frame
- Fixes the pages and records the type of frame used

2.4.2.1 Expanded Storage

Expanded storage can be thought of as an expansion of central storage. The purpose of expanded storage is to reduce the paging and swapping of pages between central storage and auxiliary storage, and thus enhance system performance. Because moving a page between central storage and expanded storage is much faster than I/O, use of expanded storage can provide a significant performance advantage.

Auxiliary Storage Manager

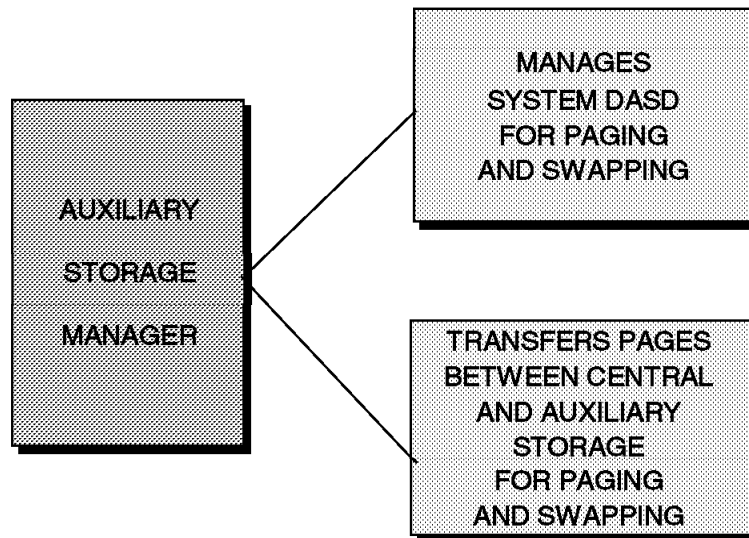


Figure 42. Auxiliary storage manager

2.4.3 Auxiliary Storage Managers

Auxiliary storage manager (ASM) is the component of the OS/390 system that is responsible for transferring virtual pages between real and auxiliary storage. This is done as either a paging operation (one page at a time) or as a swapping operation (an address space at a time). AMS manages the transfer by initiating the I/O and by maintaining tables to reflect the current status of auxiliary storage.

Enough auxiliary storage must be available for the programs and data that comprise the system. Auxiliary storage used to support basic system requirements has three logical areas:

- System data set storage area.
- Paging data sets for backup of all pageable address spaces.
- Swap data sets used for LSQA pages and private area pages that are swapped in with the address space (also called the *working set*)

For more information, see *OS/390 Initialization and Tuning Guide*, SC28-1751.

Frames, Slots and Pages

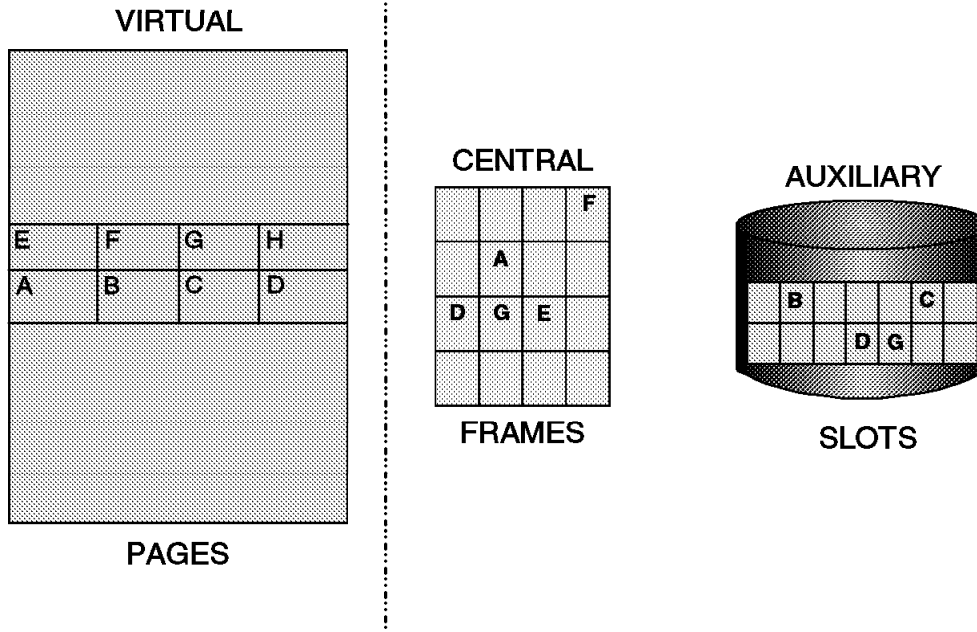


Figure 43. Frame, slots and pages

2.4.3.1 Frames, Pages and Slots

The parts of a program executing in virtual storage must be moved between real and auxiliary storage.

To allow this, OS/390 breaks the storage into blocks:

- A block of real storage is a *frame*.
- A block of virtual storage is a *page*.
- A block of virtual storage is a *slot*.

A frame, page and slot are all of the same size, 4 KB.

Paging and Swapping

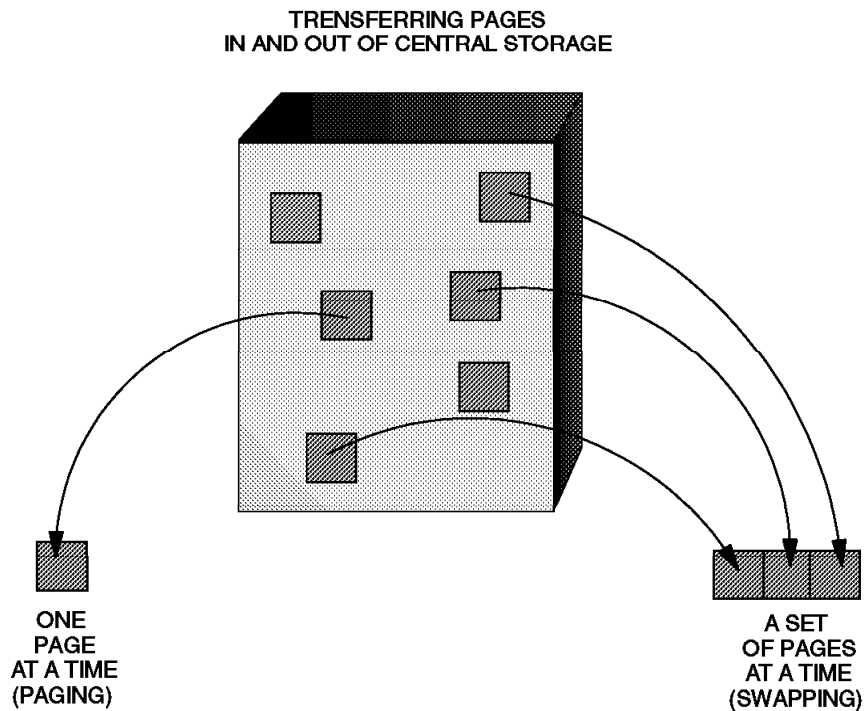


Figure 44. Paging and swapping

2.4.4 Paging and Swapping

The paging and swapping controllers of the auxiliary storage manager (ASM) attempt to maximize I/O efficiency by incorporating a set of algorithms to distribute the I/O load as evenly as is practical. In addition, every effort is made to keep the system operable in situations where a shortage of a specific type of page (swap) space exists.

RSM uses expanded storage as an extension of central storage. When a page is to be removed from central storage, RSM first considers moving it to expanded storage instead of auxiliary storage. When a page that is needed is not in central storage, RSM first checks expanded storage for the page. If the page is in expanded storage, RSM synchronously retrieves the page. If the page is not in expanded storage, RSM calls ASM to schedule asynchronously the paging I/O to retrieve the page from auxiliary storage.

When contention for expanded storage increases, the system removes pages from expanded storage to free expanded storage frames. RSM first moves the pages from expanded storage to central storage. RSM then calls ASM to schedule the paging I/O necessary to send these pages to auxiliary storage. This process is called migration. Migration completes when the pages are actually sent to auxiliary storage.

RSM is responsible for reclaiming the central storage allocated to an address space when the address space is to be swapped out of central storage. RSM is also responsible for building the control structures necessary to efficiently swap the address space back into central storage. When an address space is swapped out of central storage, RSM works with SRM to identify the working set pages that will be swapped back into central storage.

Auxiliary Data Sets

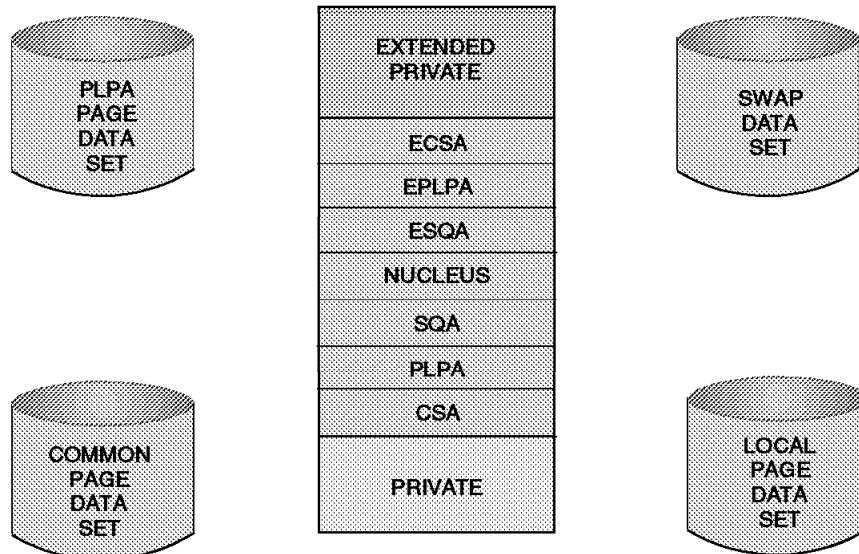


Figure 45. Paging operation

2.4.4.1 Paging

To page efficiently and expediently, ASM divides the pages of the system into classes, namely PLPA, common, and local. Contention is reduced when these classes of pages are placed on different physical devices. Although the system requires only one local page data set, performance improvement can be obtained when local page data sets are distributed on more than one device, even though some devices may be large enough to hold the entire amount of necessary page space. The PLPA and common page data sets are both required data sets, and there can be only one of each. Spillage back and forth between the PLPA and common page data sets is permissible, but, in the interest of performance, only spilling from PLPA to common should be tolerated.

The pages data sets are:

- PLPA page data set: this contains pageable link pack area. (There is only one.)
- Common page data set: this contains the non-PLPA virtual pages of the system common area. (There is only one.)
- Local page data set: this contains the private area (address space) pages, and space for any VIO data sets. Local page data sets can be dynamically added to and deleted from the paging configuration without re-IPLing the system.(There are one or more.)
- Duplex page data set: this is an optional data set, used when duplexing of the common area is requested.

Swapping Operations

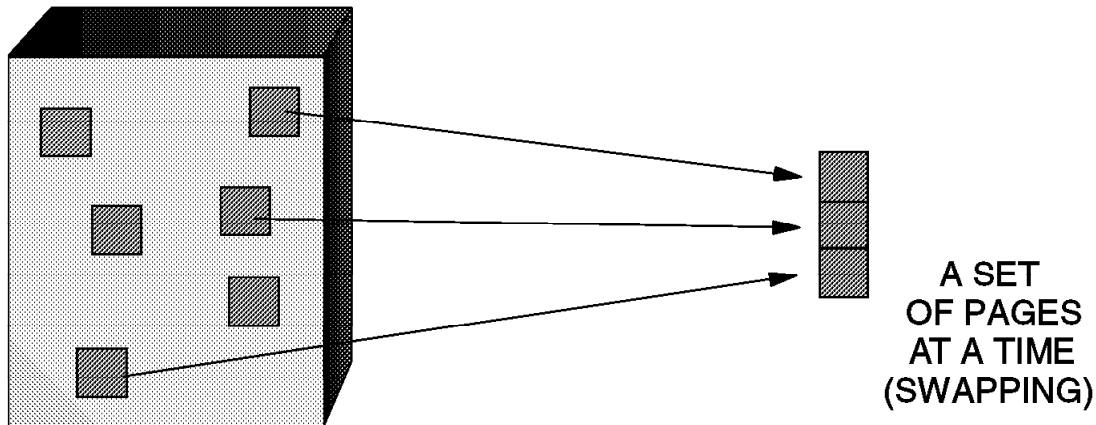


Figure 46. Swapping operation

2.4.4.2 Swapping

Swapping is the primary function used by SRM to exercise control over distribution of resources and system throughput. Using information specified by the installation through IPS and OPT parameters, and system status information that is periodically monitored, SRM determines which address spaces should have access to system resources.

In addition to the swapping controls described in the following text, SRM also provides an optional swap-in delay to limit the response time of TSO/E transactions.

There are several reasons for swapping. Some swaps are used for control of domains and the competition for resources between individual address spaces within a domain, while others provide control over system-wide performance and help increase the throughput.

ASM sends LSQA and private area working set pages to swap data sets as long as the data sets are defined and contain free space. A swap data set consists of groups of 4096-byte slots called *swap sets*. Each swap set consists of 12 contiguous slots. Swap data sets use only one seek per swap set. To ensure this seek efficiency, ASM prevents the swap set from crossing cylinder boundaries and uses the direct access device multi-track feature.

If ASM finds no swap sets, it uses a local page data set.

ASM frees swap sets immediately upon swap-in; that is, swap pages are valid on the swap data set only for that period between swap-out and swap-in.

Program - Compile, Link-edit, and Selected into Execution

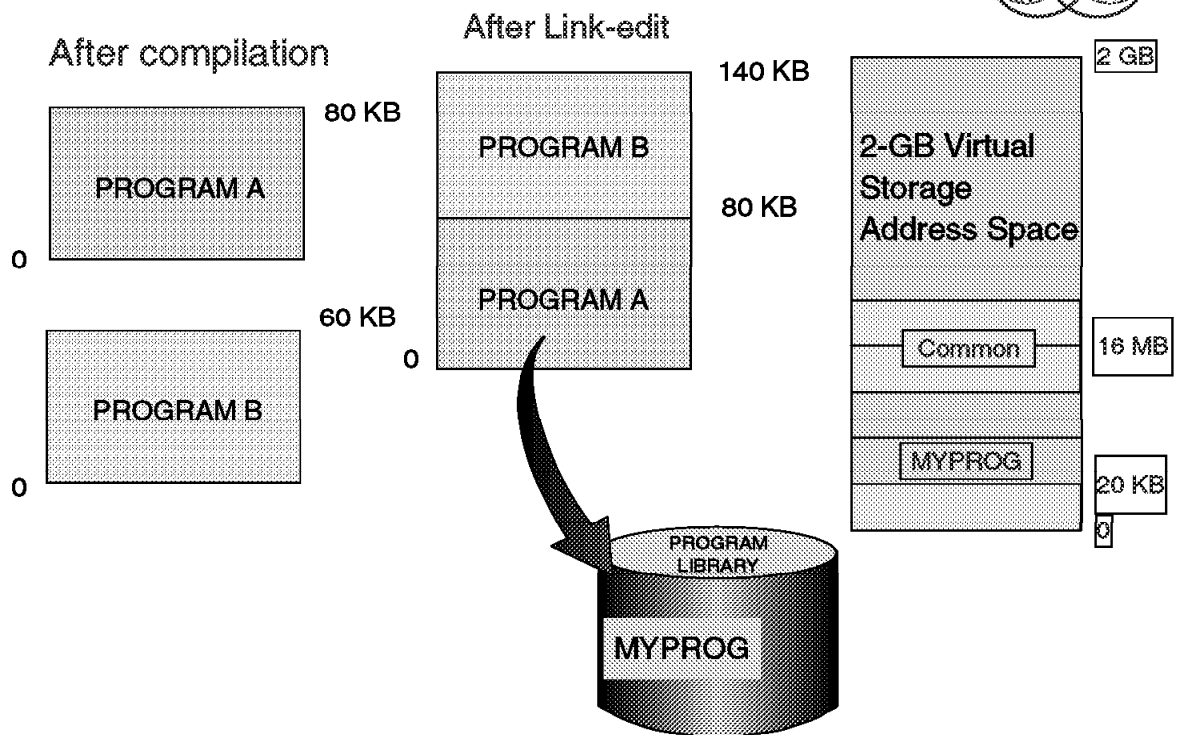


Figure 47. Program execution

2.5 Program execution

An OS/390 system may appear to be one big block of code that drives your CPU. Actually, OS/390 is a complex system comprised of many different smaller blocks of code. Each of those smaller blocks of code perform a specific function within the system.

Each system function is composed of one or more load modules. In an OS/390 environment, a load module represents the basic unit of machine-readable executable code. Load modules are created by combining one or more object modules and processing them with a link-edit utility. The link-editing of modules is a process that resolves external references and addresses. The functions on your system, therefore, are one or more object modules that have been combined and link-edited.

Chapter 3. S/390 hardware and I/O management

This chapter provides guidelines for setting up the hardware and software configuration to eliminate single points of failure. The hardware and software components provide the foundation for continuous availability, but only when the configuration is set up in a way that provides redundancy of critical elements, and eliminates inhibitors to the built-in system recovery mechanisms.

This chapter provides considerations for:

- System programmers responsible for defining the hardware configuration to OS/390
- Operators responsible for the availability of the system
- HCD definitions
- Parmlib coding

For I/O management you should understand all the tasks executed in an MVS operating system that have to do with accessing, storing, and managing for availability and performance. In a sense it includes the functions executed by the channel subsystem and the I/O control units. This chapter covers all these items.

Data Center

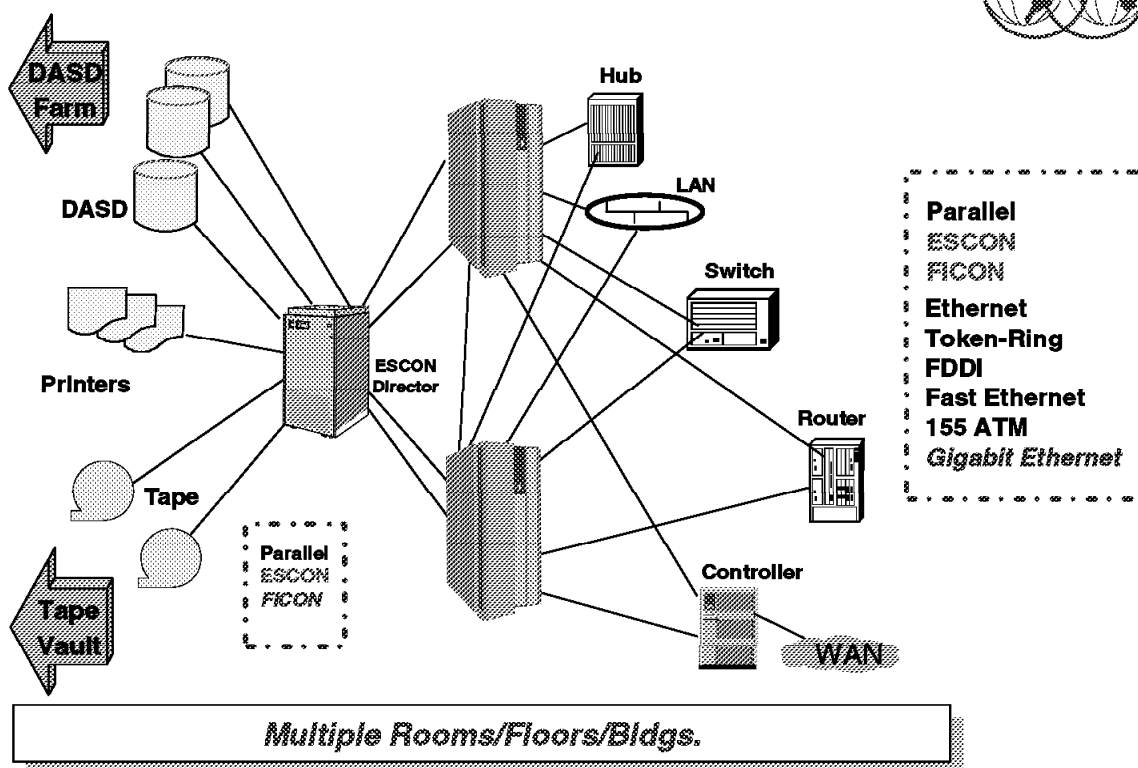


Figure 48. Data center

3.1 Data center

This visual shows a typical S/390 data center. As you can see, the complex consists of separate I/O devices and networks connected through high speed data buses to the central electronic complex (CEC), which comprises processors, processor storage, and channels. It shows connections among CECs as well. S/390 architecture provides up to 256 high speed data buses called channels or CHPIDs per CEC. The different types of channels, including the ones connecting the coupling facility (CF), a special sort of device, to be covered later in this book are:

- Parallel (S/390 I/O)
- Enterprise Systems Connection (ESCON) (S/390 I/O)
- Fiber Connection (FICON) (S/390 I/O)
- Multiprise 2003 and 3000 Internal DASD channels
- Open Systems Adapter-2 (OSA-2) (Ethernet, Fast Ethernet, Token-ring, Fiber Distributed Data Interface (FDDI), 155 Asynchronous Transfer Mode (ATM))
- OSA Express (Gb Ethernet)
- CF-Link - ISC single mode to coupling facility, which has two types: CFS (the adapter in MVS side) and CFR (the adapter in the coupling facility side)
- ICB (single mode, low distance, high rate to coupling facility), which has two types: CBS (the adapter in MVS side) and CBR (the adapter in the coupling facility side)

In order to communicate with the I/O devices attached to the channels, both the operating system (MVS) and the channel subsystem (formed by the System Assisted Processor (SAP) and channels) need to know the I/O configuration, which contains the following types of information:

- What devices are attached to each channel
- The device types
- The device addresses: subchannel number, device number, and device address (also called unit address)
- What protocols to use
- Channel type

This information is provided by the installation and is mainly stored in unit control blocks (UCB) for MVS and unit control words (UCW) for the channel subsystem. The hardware configuration definition facility (HCD) is an MVS component used to create or update the I/O configuration.

9672 G6 System Structure

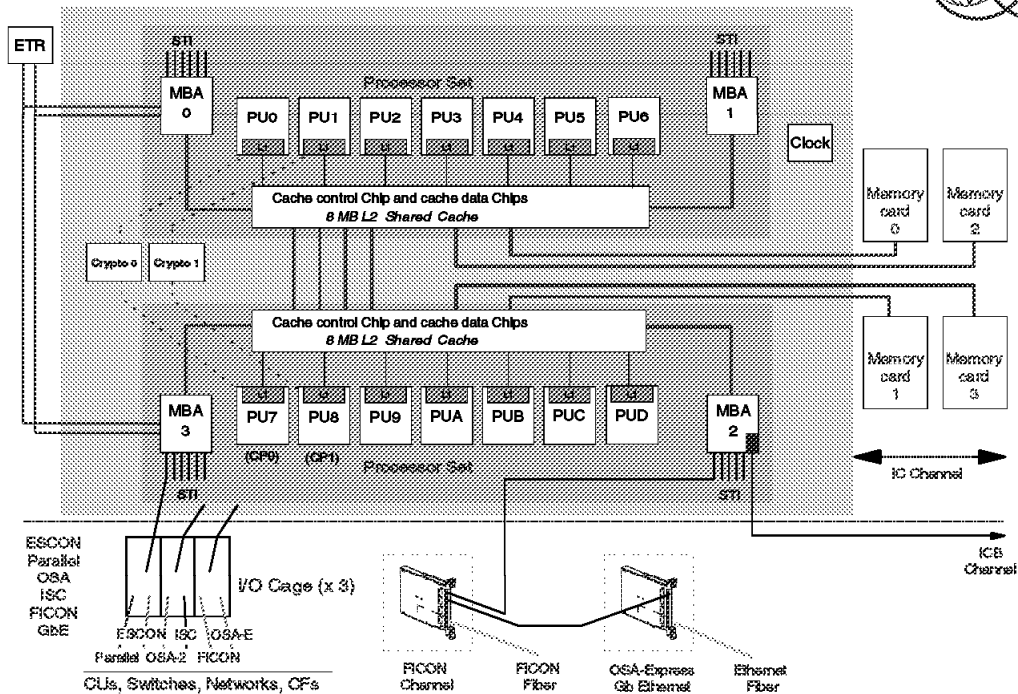


Figure 49. Logical processor structure

3.2 Logical processor structure

This visual shows the logical design for the 9672 G6 components. The design is bi-nodal, with two identical sets of processors. Each of the two sets contains:

- Seven PUs (dual execution units in each) with L1 cache sharing L2 cache.
- 8 MB L2 cache (four chips) for data and instructions.
- Two memory cards.
- Two Memory Bus Adapters (MBAs) connecting six self-timed interconnects (STIs) each.
- 12 STI busses (data pipes) to gather or send I/O data. The STI bandwidth of 333 MB/sec bidirectional allows I/O data to be moved very fast. The STI busses connect MBA with the ESCON, Parallel, OSA, ISC, FICON, and ICB channels; then, through the channels, MBA connects to the CUs, switches, networks, and CFs.
- FICON and OSA-Express are connected to STI in a daisy chain topology. ICB has a dedicated STI, and for the other channel types, each STI may serve four channels.

Channel Subsystem Logic

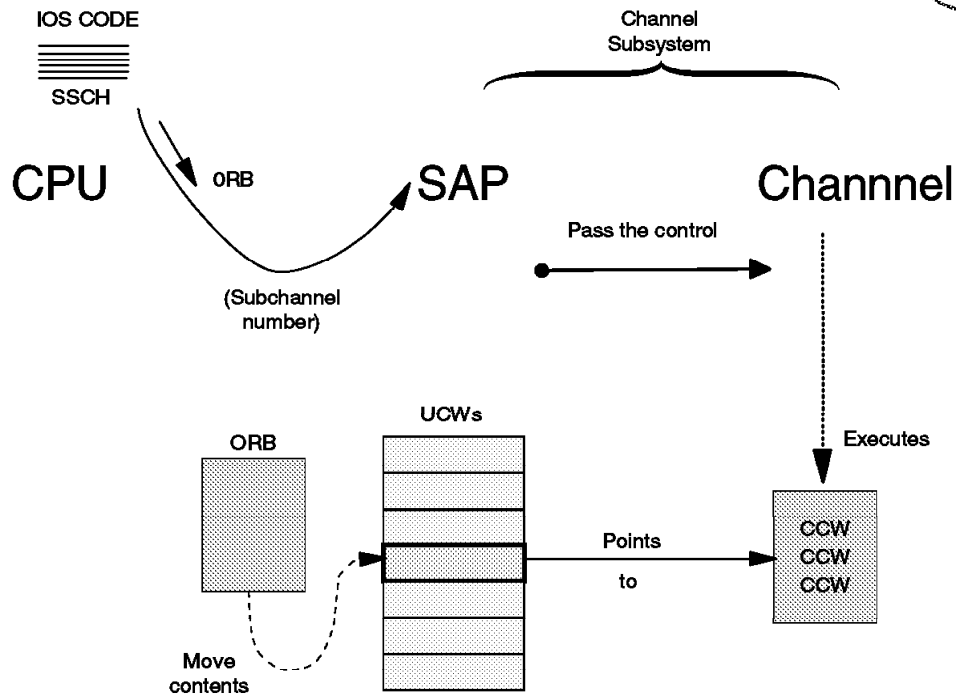


Figure 50. Channel subsystem logic

3.3 Channel subsystem

The S/390 channel subsystem contains:

- A special processor unit (PU) called an SAP. A PU is a 9672 processor. SAP is a PU that runs I/O licensed internal code (LIC). SAP alleviates the CPU involvement during the execution of an I/O operation. It schedules an I/O operation, but is not in charge of the movement between central storage (CS) and the channel. SAPs also control data movement between CS and expanded storage (ES), when the Asynchronous Data Mover Facility (ADMF) is used. ADMF is a fast way to move data between these two memories (used by DB2 for hiperpool support).
- Channels, which are special processors able to communicate with I/O control units and manage the movement of data between central storage and these control units. Being more specific, the channels can:
 - Send channel commands from the processor to a CU via electrical signals
 - Transfer data during read and write operations
 - Receive status at the end of operations
 - Receive sense information from control units

An I/O operation starts when a Start Subchannel (SSCH) instruction is executed by the Input Output Supervisor (IOS), an MVS component, which issues the instruction on behalf of an MVS process. It ends when an I/O interrupt is received by the CPU (forcing the execution of IOS code again).

3.3.1 Start Subchannel (SSCH) logic

The Start Subchannel (SSCH) is a privileged instruction issued by the Input Output Supervisor (IOS), an MVS component, used to start an I/O operation. SSCH has two operands:

- Subchannel number, which is an index to the UCW associated with the I/O device
- Operation Request Block (ORB) address. The ORB contains information about *what do to* during the I/O operation; among other fields it contains the channel program address.

The SSCH microcode moves the ORB contents into the respective UCW and places the UCW in a specific Hardware System Area (HSA) queue named the *initiative queue*. After that process completes, the next IOS instruction is executed, allowing the use of the CPU in another task. HSA is a piece of central storage not addressable by MVS. It is allocated at power-on reset (POR) and contains microcode work areas, and the I/O configuration (UCWs are stored in HSA) used by the channel subsystem.

3.3.2 SAP logic

The SAP finds the UCW in the initiative queue and tries to find a channel that succeeds in *initial selection* (connects to a control unit and starts the I/O operation). Initial selection may be delayed if:

- All channel paths (serving the device) are busy.
- The ESCON director port is busy.
- The control unit interface is busy.
- The device is busy, due to shared DASD activity or a previous not ended Write Format operation.

During all of these delays, the request is serviced by SAP without MVS awareness. When the I/O operation finishes (device-end status is presented) SAP queues the UCW (containing all the I/O operation final status) in the I/O interrupt queue.

SAP uses the I/O configuration information described in the Hardware Configuration Definition (HCD) to determine which channels and control units can reach a target device.

3.3.3 Interrupt processing

S/390 is an I/O interrupt-driven architecture, as opposed to a polling mechanism. The only exception to this rule is the handling of the MVS/Coupling Facility communication, where polling is used.

When a CPU I/O interrupt is enabled and the CPU detects the UCW in the interrupt queue, the I/O interrupt is accepted and the I/O PSWs are asynchronously swapped passing the control back to IOS. An Interrupt Response Block (IRB) is moved to storage describing the final status of the I/O operation. Another way to receive this interrupt is by IOS synchronously issuing the test pending interrupt (TPI) instruction.

Usually, IOS then posts the MVS process waiting for the I/O operation.

In certain error situations the I/O interrupt is not generated within an expected time frame, then the MVS component Missing Interrupt Handler (MIH), a timer-driven routine, alerts IOS about this condition.

Multiple Paths to a Device



```

IEE459I 11.48.20 DEVSERV PATHS 667
UNIT DTYPE M CNT VOLSER CHPID=PATH STATUS
      RTYPE SSID CFW TC   DFW  PIN  DC-STATE CCA  DDC  ALT  CU-TYPE
OCC0,33902 ,0,000,PMUDAS,10=+ B0=+ 88=+ DC=<
      000A Y YY.  YY.  N  SIMPLEX  00  00          3990-6
***** SYMBOL DEFINITIONS *****
O = ONLINE                      + = PATH AVAILABLE
< = PHYSICALLY UNAVAILABLE
  
```

```

IEE174I 11.52.23 DISPLAY M 670
DEVICE OCC0  STATUS=ONLINE
CHP          10 B0 88 DC
PATH ONLINE  Y  Y  Y  N
CHP PHYSICALLY ONLINE Y  Y  Y  Y
PATH OPERATIONAL Y  Y  Y  N
  
```

Chpid DC is unavailable to dev CC0
but there is still access to the CC0
device through the other 3 paths

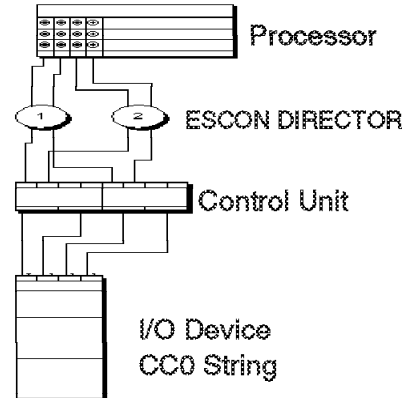


Figure 51. Multiple paths to a device

3.4 Multiple paths to a device

I/O devices are attached through control units to the channel subsystem. A path is simply a route to a device.

Control units may be attached to the channel subsystem via more than one channel path, and an I/O device may be attached to more than one control unit (if the cache is common as in the two halves of a 9390).

An I/O device may be accessed by an MVS image through as many as eight different channel paths. Also an I/O device may be accessible to a CEC (channel subsystem) by as many as eight different channel paths.

The total number of channel paths provided by a channel subsystem depends on the model and the configuration; the maximum number of channel paths is 256 per CEC.

Device Number

- ★ External device label
- ★ Four hex digits in range 0000-FFFF
- ★ Arbitrarily assigned by sysprog
- ★ Used in JCL, commands, messages, EREP

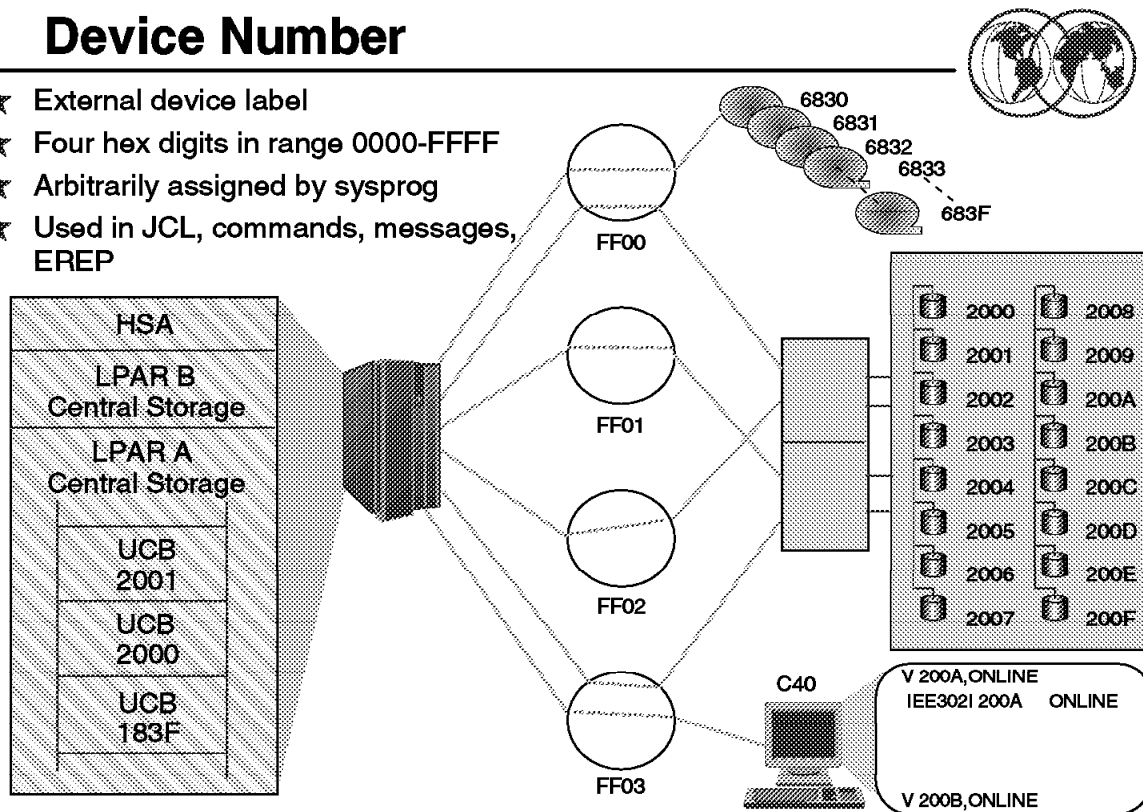


Figure 52. Device number

3.5 Device number

The device number is a kind of nickname for the device. It is assigned at HCD creation by the installation. It is used to reference the device during the communication by MVS and human beings, as well as in messages and console commands, JCL, RMF device reports, and at IPL to indicate the SYSRES device.

A device number is contained in a 16-bit value field, whose valid range is 0000-FFFF, which allows for a maximum of 65,536 devices. The device number is stored in the UCW (HSA) at power-on reset (POR) and in UCB at IPL. UCW represents the device from the channel subsystem point of view; a UCB represents the device from the IOS point of view.

Subchannel Number



- ★ Device identification used for communication
 - ▶ Between CPU and Channel Subsystem
- ★ Two bytes
 - ▶ Values from 0000 to FFFF (64K devices)
- ★ Index for the UCW in the UCW table
- ★ Used by SSCH microcode
 - ▶ To find the requested UCW
- ★ Stored in the UCB at IPL

Figure 53. Subchannel number

3.6 Subchannel number

The subchannel number is another way for addressing an I/O device. It is a value that is assigned to the subchannel (UCW) representing the device at POR time. It indicates the relative position of the UCW in the UCW table. The specific value depends on the position of the statement IODEVICE in the IOCP.

It is used to reference the device during the communication between MVS and the channel subsystem as well as during the SSCH instruction and the interrupt processing. The subchannel number was designed to speed up the search of a UCW during the SSCH processing. The same device accessed by different logical partitions has one UCW per image. In a G6 the maximum number of UCWs is 80,000.

The subchannel number is expressed in a 16-bit value, whose valid range is 0000-FFFF, allowing for a maximum of 65,536 devices per MVS image. Stored in the UCB at IPL, it is not declared during HCD initialization.

Subchannel Number II



- Channel Subsystem device label
- Four hex digits in range 0000-FFFF
- Assigned during IOCP
- Independent of device number
- Used between MVS<->CSS

ACTIVATE IODF=55
DYNAMIC ACTIVATION IN
PROGRESS

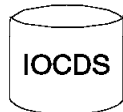
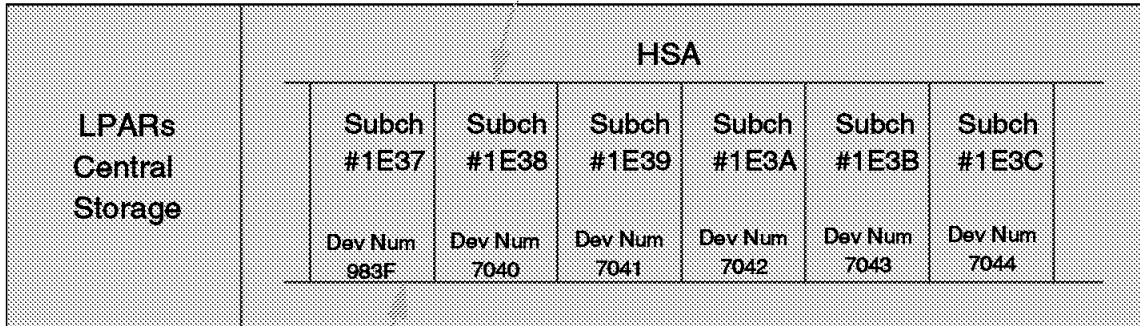
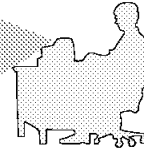


Figure 54. Subchannel number II

3.6.1 Subchannel number II

This visual shows the offset concept of the subchannel number in the UCW table.

Unit Address



- ★ Two hex digits in range 00-FF
- ★ Used to select a device on a channel
- ★ Assigned by sysprog in HCD device definition

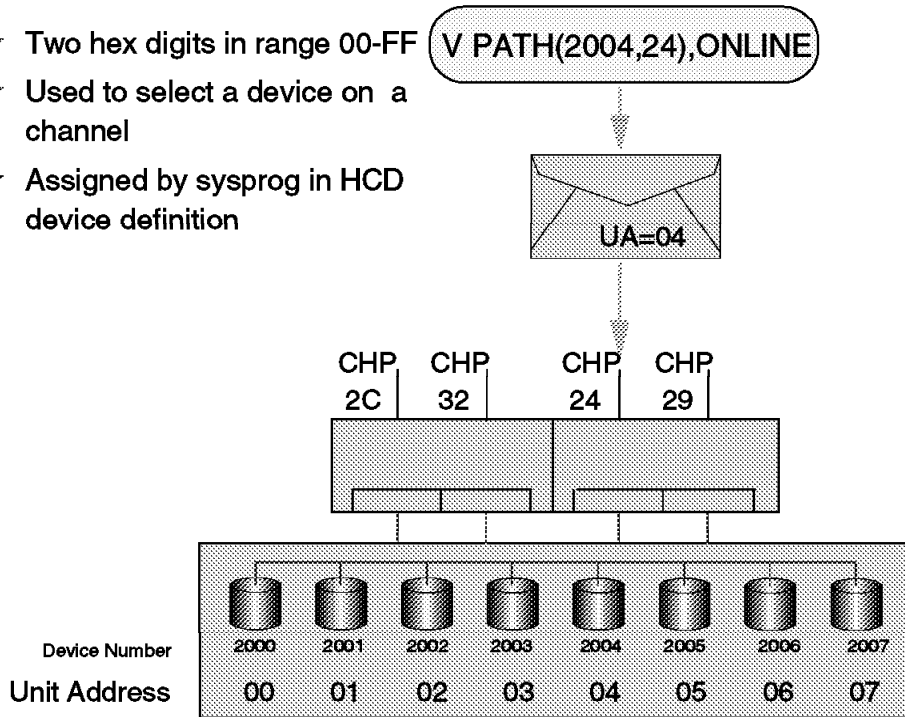


Figure 55. Unit address

3.7 Unit address

The unit address (UA) or device address is used to reference the device during the communication between a channel and the control unit serving the device. The UA is two-hex digits in the range 00-to-FF stored in the UCW as declared in HCD, and it is transmitted over the I/O interface to the control unit by the channel.

Each parallel channel supports the full unit address range of 00-to-FF and so allows connection of up to 256 devices. For an ESCON interface, architecturally the control unit supports the full unit-address range of 00-to-FF and may support up to 256 devices.

The UA has no relationship to the device number or the subchannel number for a device, but care must be taken during device definition since in HCD the unit address defaults to the last two digits of the device number, if not explicitly specified. Some control units (3990 control units for example) attached to S/390 ESCON serial interfaces require a unit address range starting at X'00'.

The unit address defined in HCD must match the unit address on the control unit where the device has been physically installed by the hardware service representative.

In the example, the operator issues the V PATH command for device number 2004. A separate I/O operation must be started for each device. The device number 2004 corresponds to the UA 04 and this is the address that the channel sends to the control unit.

In summary, the device number is used in communication between an operator and MVS; the subchannel number is used in communication between MVS and channel subsystem; and the unit address is used in communication between the channel and the control unit.

Control Unit Function



- ★ Every channel connects to a control unit
- ★ Control unit may located
 - ▶ In the CPU (CTC or OSA)
 - ▶ In the device (6262)
 - ▶ Or in a separate unit (3990)

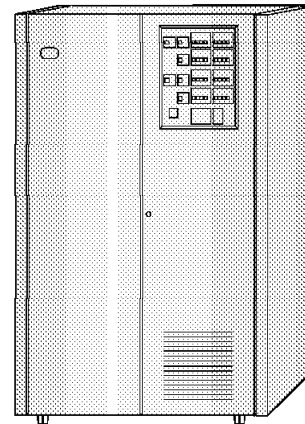


Figure 56. Control unit function

3.8 Control unit function

All I/O channels connect to a control unit prior to connecting to the I/O device. The role of the control unit is to regulate the I/O-device operation. It provides the caching and buffering capabilities necessary to operate the associated I/O device. It also does error recovery. Today's control units are very complex and sometimes are called I/O subsystems. From the programming point of view, most control unit functions merge with I/O device functions.

The control unit function may be:

- Integrated with the I/O device
- In the CPU
- In a separate control unit

Parallel Channel



★ Multi-dropped or daisy-chained environment

★ Two copper cables (40 core wire) Bus and Tag

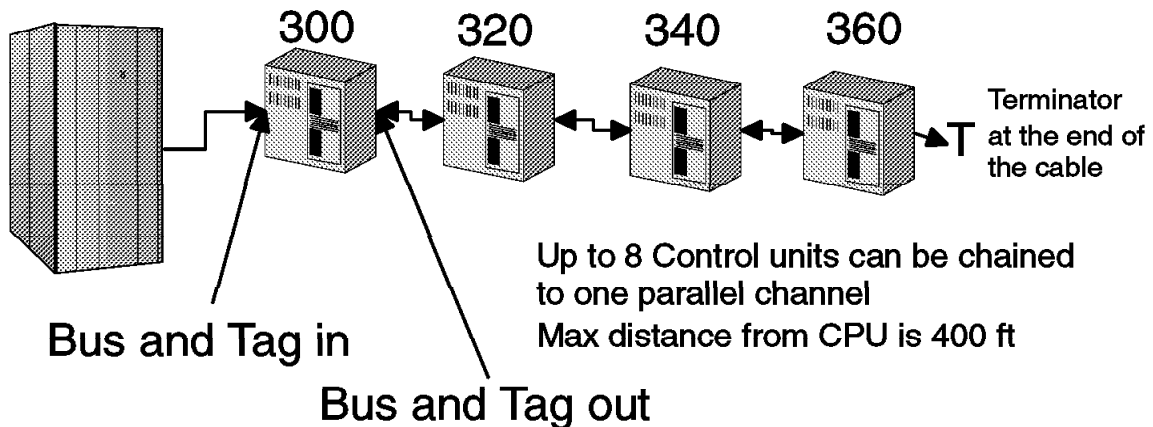
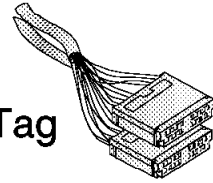


Figure 57. Parallel channel

3.9 Parallel channel

A parallel channel is the communication protocol and media specification link between a central electronic complex (CEC) and an I/O control unit (CU) with a parallel interface. It is designed for S/370 or ESA/390 modes. There are two types of parallel channels: byte multiplexor and block multiplexor.

Parallel channels use the parallel S/370 or ESA/390 I/O interface which defines two cables, called bus-and-tag cables. A bus cable carries information (one byte each way), and a tag cable indicates the meaning of the information on the bus cable. Bus and tag cables are connected sequentially to the control units (one after the other control unit). This is often referred to as *daisy chaining*. The last control unit on the string is equipped with terminator blocks.

The architectural limit to the number of control units in the chain is 256, but due to physical limitations, it is restricted to a maximum of eight control units on a channel.

Daisy chaining better utilizes a channel for slow control units, but a single failing control unit or a bad cable connection can influence other control units chained on the channel. As only one data transfer can occur up or down a given channel at any one time, daisy chained control units can cause contention on the channel which can slow down I/O operations and degraded performance. Up to 256 I/O devices can be addressed on a single parallel-I/O interface.

The maximum length of a parallel channel interface is restricted to 400 feet from the processor to the last control unit in the chain. This is due to electrical skew on the cable.

ESCON Channel



- ★ Fiber optic
- ★ Point-to-point architecture
- ★ Communicates from the processors to peripherals
- ★ Supports multiple path connections to peripherals
- ★ Provides CTC processor-to-processor communication
- ★ Provides routing information
- ★ Detects media connection and disruption
- ★ Identifies and verifies paths
- ★ Assists in error recovery

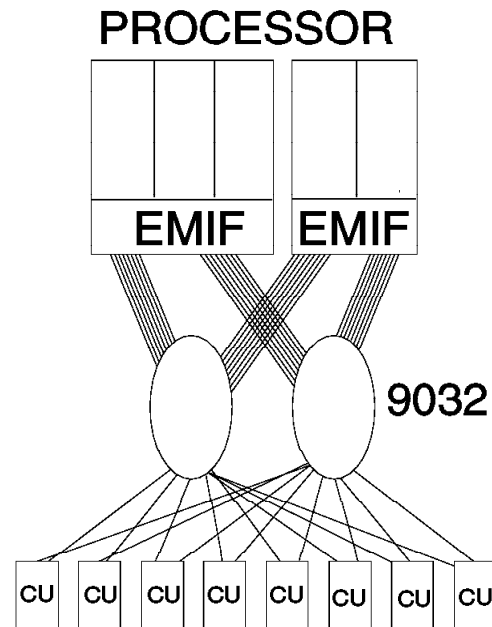


Figure 58. ESCON channel

3.10 ESCON channel

An ESCON channel, in a general sense, performs the same functions as a parallel channel, as described in 3.9, “Parallel channel” on page 86.

Going into more detail, we may say that ESCON is essentially a point-to-point (or one to one) architecture, which establishes a *serial* protocol and media specification for the communication among channels and control units. The ESCON channel implementation uses fiber optic technology, ideally suited for high-speed operation over long distances.

Optical technology is less susceptible to errors caused by electrical noise. Optics also have very low radiation properties, which make them more secure than electrical cables.

An ESCON channel is made of three pieces:

- ESCON channel itself (a card in the CEC)
- An ESCON port; that is, a connector
- An ESCON link—an optical cable

The point-to-point architecture lets you route data using dynamic switches.

These switches are called 9032 ESCON Directors (ESCD). They can be used to connect multiple channels to the same control unit, or multiple control units to the same channel. Also, it allows connecting channels to other channels, and control units to other control units. ESCDs allow longer

distances and high flexibility in an I/O reconfiguration, mainly in a failure recovery situation. The directors can be controlled and managed from host-based functions such as ESCON Manager and HCD. Host systems (when controlling ESCDs) and the ESCDs communicate using channel connections assigned to specific unit addresses in the ESCD.

ESCON provides bidirectional (not concurrently) serial-bit transmission, in which the communication protocol is implemented through sequences of special characters and through formatted and architected defined sets of characters. A sequence is a set of characters in a predefined order used to signal specific states or transition to states, such as a port entering offline state. The ESCON I/O interface defines two types of frames, one to control the link and associated elements, and another to control device operation. Each frame contains addressing information that uniquely identifies the sender and the receiver.

As stated before, the transmission medium for the ESCON I/O interface is a fiber optic cable. Physically, it is a pair of optical fibers that provide two dedicated unidirectional serial bit transmission lines. Information in a single optical fiber always flows in the same direction. Thus, one optical fiber is used to receive data while the other is used to transmit data.

ESCON Multiple Image Facility (EMIF) allows the same physical channel to be shared among multiple images from an LPAR.

The ESCON architecture of the S/390 channel subsystem provides:

- Communication from the processors to peripherals
- Multiple path connections to peripherals
- Processor-to-processor communication using serial channel-to-channel connections
- Routing information
- Detection of media connection and disruption
- Identification and verification of paths
- Assists for error recovery
- Support for logical channels
- A logical path to devices

S/390 ESCON Concepts

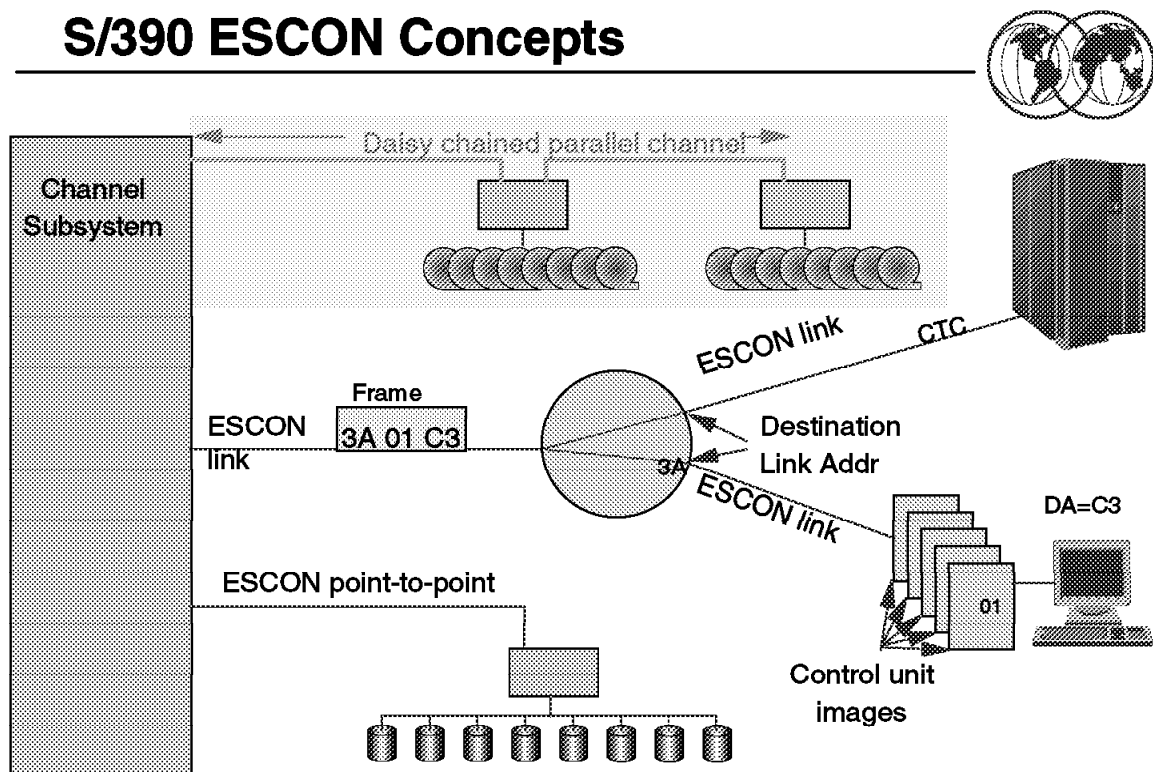


Figure 59. S/390 ESCON concepts

3.11 S/390 ESCON concepts

ESCON and the S/390 architecture introduce extensions to the device addressing scheme described previously, however, most of the changes are transparent to the application program. The S/390 ESCON changes are made in support of the new channel interfaces, and essentially, the interface between MVS and the channel subsystem remains the same as for the old S/370 XA architecture.

This visual shows the changes introduced in the ESCON device path structure. The daisy-chained parallel channel connecting to two different control units is shown for comparison.

3.11.1 ESCON link

An ESCON channel may connect either directly to an ESCON-capable control unit, called a point-to-point connection, or connect to an ESCON-capable control unit through an ESCON director, called a switched connection.

The ESCON environment provides increased connectivity by providing for each ESCON channel to connect to up to 256 links. Each link may attach to a control unit. This visual shows one ESCON channel connecting to two different ESCON links and hence two different physical control units. One of them is a CTC (connected to another CEC) and the other refers to a control unit with different images.

3.11.2 Link address

The link address is a two digit hex number, in the range 01-to-FE, that identifies the sender and the receiver during the communication between an ESCON channel and its control units. In other words, link addresses are used in the ESCON frame to identify both destination and source link-addresses. Link addresses 00 and FF are reserved by hardware.

Each port in a ESCD has a port number from 01-FF.

The link address (when the channel is connected to a ESCD) corresponds to the ESCON director port number where the ESCON link attaches. If the channel is not connected to an ESCD port the link address is FE.

Imagine for example, a channel (sender) starting a conversation with a control unit (receiver), sending an ESCON frame through an ESCD. The receiver address is made up of the following elements:

- *Link address* associated with the ESCD port number of the control unit (passed by the installation in HCD). This information is used by the ESCD to route a frame originating in the channel port to the appropriate link port where the physical control unit is connected. The link address value in the visual is 3A.
- *Control unit image* (CUADD value). Some physical control units that support single-path devices, such as the ESCON capable 3174 models 12L and 22L, allow concurrent connections to multiple hosts through the same physical interface, or control unit adapter. This support is provided through the use of different Control Unit Images (CUI). As with the link address, the CUI, also known as the control unit address (cuadd), forms part of the device selection scheme in an ESCON environment. The control unit image value in the visual is 01.
- *Device unit address* The device address in the visual is C3.

The sender (or source) address indicates the:

- *Channel link address* (port number in the ESCD), this information is not provided by the HCD, the channel obtains this value from the ESCD at initialization.
- *LPAR system image ID*, to allow the implementation of EMIF.

When responding to the channel, the control unit swaps the receiver address and the sender address in the new frame.

There are no MVS commands available to display the ESCON components of a path to a device. Only ESCON Manager commands can display the link addresses used by channels supporting the paths to a device.

Mapping Device Number in Unit Address

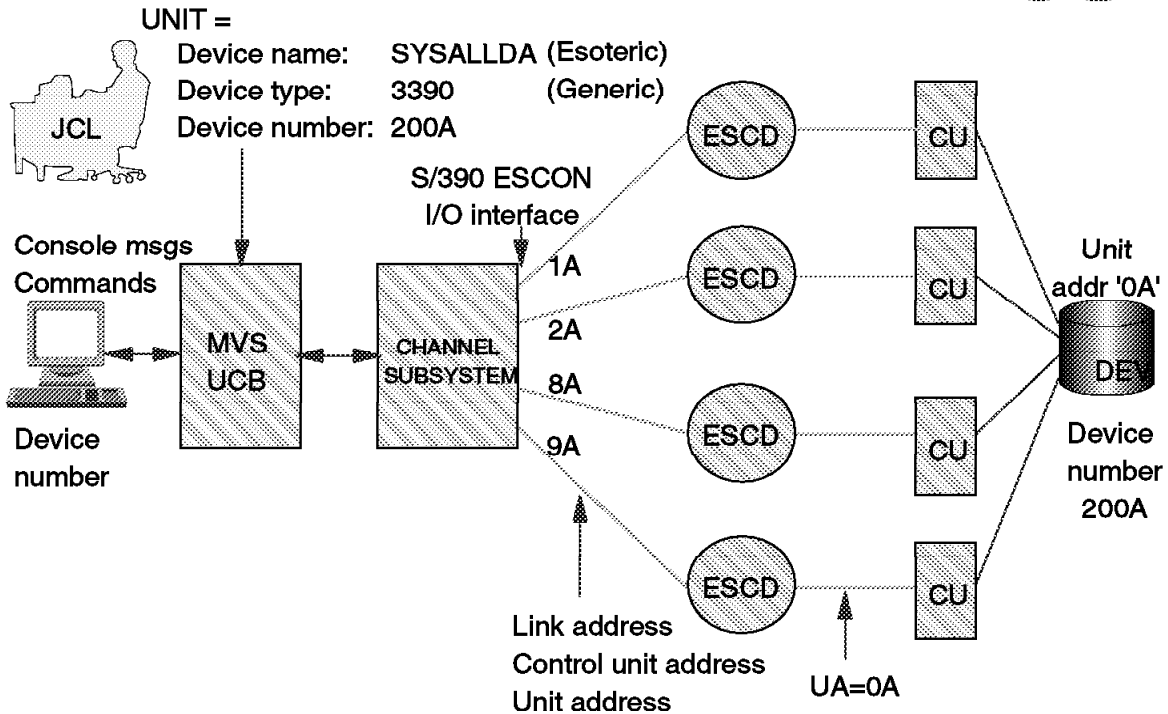


Figure 60. Mapping device number in unit address

3.12 Mapping device number in unit address

This visual shows how in an ESCON environment the device number is mapped to the device address.

The UNIT parameter in the DD card specifies the device number through an esoteric or generic name. Also, a specific device number may be specified. This information is used to allocate the data set associated with this DD card.

Allocation of a data set in MVS terms means to associate the data set DD card (through a TIOT, a control block) with the UCB of the device containing the data set. The UCB has all the information needed for the preparation of the SSCH instruction by IOS, including the subchannel number.

The channel subsystem, through the SAP, will find in the UCW (through the subchannel number) the receiver address (link, control unit, and unit address) to be placed in the ESCON frame.

The visual also shows that any operator command that refers to a device uses the device number.

ESCON CUADD Function

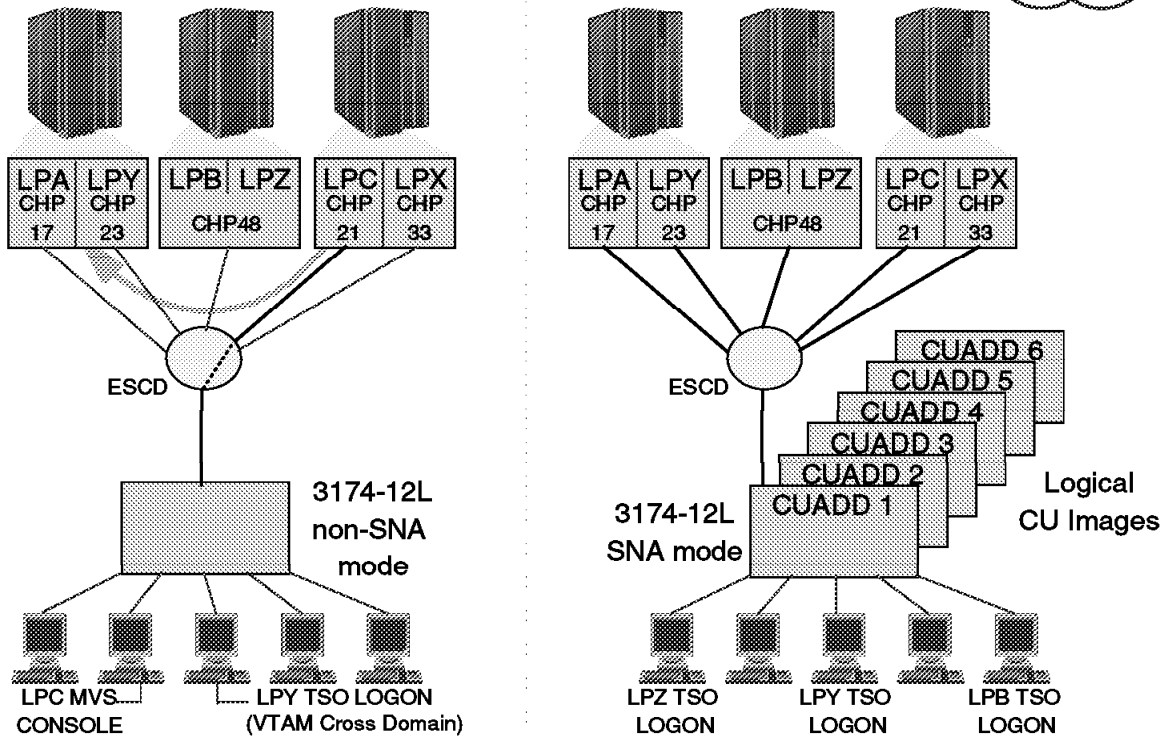


Figure 61. ESCON CUADD function

3.13 ESCON CUADD function

The ESCON I/O interface allows physical control units to define multiple images that can be addressed separately. These control unit images can control a shared or independent set of devices. The control unit image can be considered to operate independently from all other control unit images, for example, each can establish communication with its own channels.

The ESCON models of the 3174 control unit can be configured in either non-SNA or SNA mode. The visual shows that when the 3174 is configured in non-SNA mode, it operates in single-host mode. That is, it supports only one logical control unit image. Only one host can establish an ESCON logical path with the control unit.

When the 3174 is configured in SNA mode, it operates in multi-host mode. It supports up to eight logical control unit images (do not mix this concept with the logical control unit in channel subsystem) and therefore up to eight hosts can establish an ESCON logical path with the control unit. Each display can have from one-to-five sessions. Host session selection is managed by the jump key of the displays attached to the 3174. Each host must establish an ESCON logical path with the 3174, and the host session switching is then managed at the ESCON director.

MVS supports a console (NIP or MCS) attached to an ESCON 3174, but only when it is configured in non-SNA mode. Therefore, it is not possible to share a console device between more than one MVS image.

As you can see the 3174 used the CUADD feature to enable single path devices to connect to multiple hosts.

The RAMAC Virtual Array also uses the CUADD function, but for a different reason. It uses CUADD to enable its 256 device address to be converted into four CUADD statements with 64 address defined in each. The effect of the CUADD function in this scenario is to allow the control unit to address more devices on a channel.

Another use of CUADD is to implement EMIF CTC. Refer to 3.16, "ESCON channel-to-channel (CTC)" on page 97 for more information about CTC ESCON.

ESCON Director (Switch)



★ Terminology

- ▶ Matrix
- ▶ Blocked Port
- ▶ Prohibited Port
- ▶ Allowed Port
- ▶ Chained Connection
- ▶ Dedicated Port

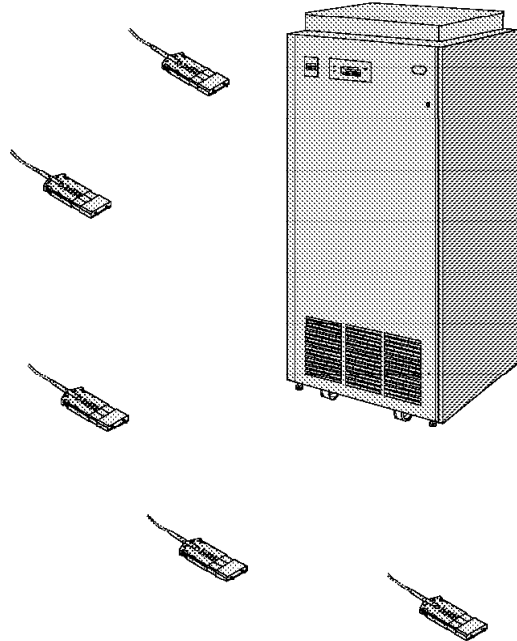


Figure 62. ESCON director (switch)

3.14 ESCON director (switch)

To implement the switching functions of ESCON, a new class of product was introduced. This product, called ESCON director (ESCD), dynamically connects channels and control units only for the duration of an I/O operation. The connection is dynamic because it is only held during the conversation. After that the same port can be used to connect to a different control unit. ESCDs do not introduce delays and can switch millions of connections per second. ESCDs are the centerpiece of the ESCON topology.

ESCDs switch connections between ports under the direction of the link address as provided by the channel and the attached ESCON control units within the frames.

Because the ESCD redrives the light signal, it can be used as a channel extender, to communicate over long distances.

Apart from dynamic switching, the ESCD can also be used for static switching (also called dedicated), where a port is always connected to another port. When a channel and a control unit are connected through two ESCDs (for longer distance purposes) one of the connections must be static. The reason is that in the frame there is just one sender link address and just one receiver link address.

In order to store configurations and handle errors, for example, the director has an integrated control unit, addressed by the host like any other control unit. The channel subsystem can communicate with the ESCD just like any other control unit in the ESCON subsystem. The director dynamically switches I/O requests for itself to its own internal port.

The director consists of multiple bidirectional ports to which channels and control units may be attached. The ports are capable of any-to-any connection, but the installation may restrict this access by:

- Blocking ports (allowing no access at all)
- Prohibiting port-to-port connections
- Dedicating port-to-port connections
- Allowing connections (opposite of prohibiting)

These restrictions can be used for example, to prevent a test MVS system from accessing the production DASD, but if a disaster happens, automation acting upon the port's state may reverse the situation.

The port configuration is held in a switch port matrix in the ESCON director. The port matrix can be read and written from an attached host using the ESCON Manager. An initial port matrix is held on the disk of a PC directly-attached to the director; however, this disk data is not directly addressable by the ESCON Manager and hence cannot be changed remotely.

HCD uses the ESCON Manager to communicate with ESCON directors, and can effect changes only through the ESCON Manager. There are two types of ESCDs:

- *9032 ESCD* has from 28 to 60 external ports (in four-port increments). Each 9032 port provides for the attachment of any ESCON channel, ESCON extended-distance channel, control unit, 9034 or 9035 ESCON Converter, or another ESCD. The 9032 model 5 may contain the FICON bridge card.
- *9033 ESCD* has from 8 to 16 external ports. Each 9033 port provides for the attachment of any ESCON channel or control unit, 9034 or 9035 ESCON Converter, or another ESCD.

ESCON Director Matrix



```

Switch ID . . . . . : AA          Switch AA
Switch configuration ID . : ESCD001A  Default connection : Allow

      Ded      --Dynamic Connection Ports Cx--
/ Port Name +   B Con + 0 1 2 3 4 5 6 7 8 9 A B C D E F
- C0 TO_3990XA0_'E'   N ___ \ P P P P P P P P P P P P P P P
- C1 FROM_982A_CHPID_64 N ___ P \ P P P P P P P P P P P P P
- C2 FROM_982B_CHPID_B0 N ___ P P \ P P P P P P P P P P P P
- C3 TO_9034_P6       N F7  P P P \ P P P P P P P P P P P P
- C4 TO_3990XB0_'E'   N ___ P P P P \ P P P P P P P P P P P
- C5 TO_3172-MOD3_OSL N ___ P P P P P \ P P P P P P P P P P P
- C6 FROM_982B_CHPID_B1 N ___ P P P P P P \ P P P P P P P P P
- C7 FROM_982A_CHPID_6C N ___ P P P P P P P \ P P P P P P P P
- C8 TO_3990XC0_'E'   N ___ P P P P P P P P \ P P P P P P P
- C9 FROM_982A_CHPID_65 N ___ P P P P P P P P P \ P P P P P P
- CA FROM_982B_CHPID_B8 N ___ P P P P P P P P P P \ P P P P P
- CB TO_9034-AT2A     N E9  P P P P P P P P P P P \ P P P P
- CC TO_3990XD0_'E'   N ___ P P P P P P P P P P P P \ P P P
- CD TO_OSL_SP-2      N ___ P P P P P P P P P P P P P \ P P
- CE FROM_982B_CHPID_B9 N ___ P P P P P P P P P P P P P \ P
- CF TO_OSL_3172-3    N ___ P P P P P P P P P P P P P P P \
  
```

Dedicated connection
from port C3 to port F7

Allowed connection from
port C2 to port CF

Figure 63. ESCON director matrix

3.15 ESCON director matrix

This foil shows an example of an ESCON switch matrix. The ESCON switch matrix can be stored in a number of places:

- The active switch data is in the ESCD itself.
- The switch may have a copy of the active configuration on its support console disks.
- The ESCON Manager may have a copy of the active configuration in its working storage, being worked on as an ISPF table.
- The ESCON Manager may have saved copies of possible configurations in ISPF tables.
- HCD may have configurations stored in the active production IODF.
- HCD may have configurations stored in other production IODFs.
- HCD may have configurations stored in work IODFs.

The recommendation for matrix configuration is to protect all ports and only allow access where required. This simplifies I/O problem determination and reduces overhead in event notification.

ESCON CTC

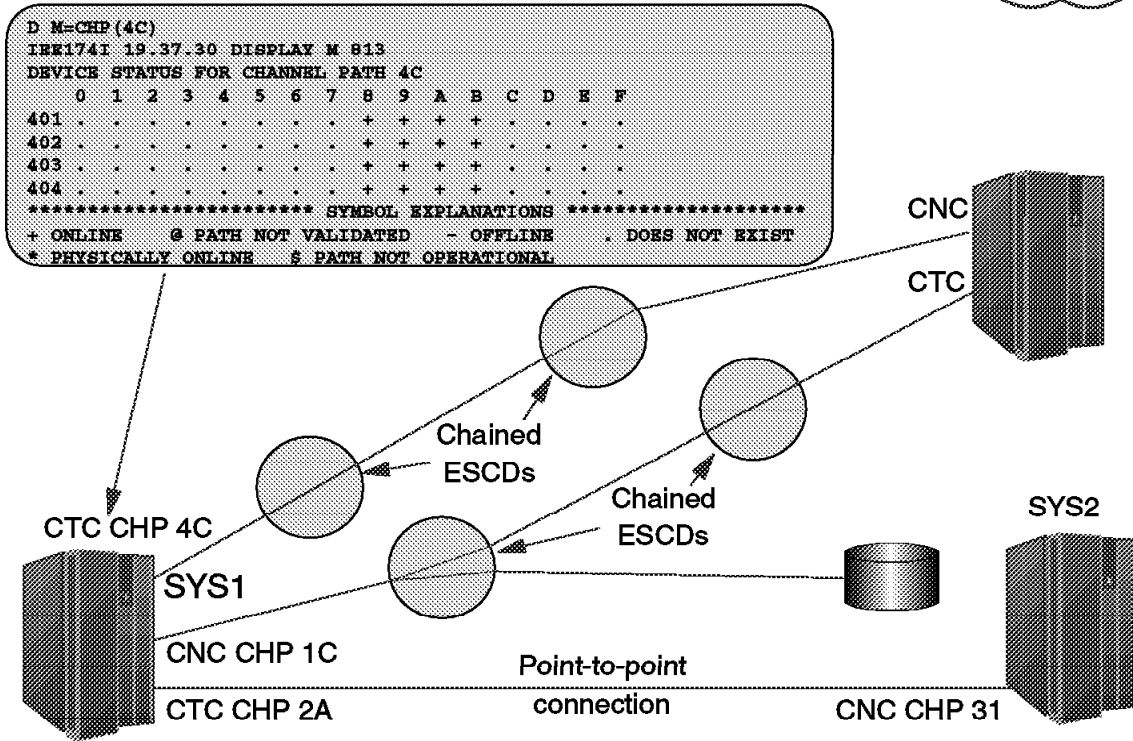


Figure 64. ESCON CTC

3.16 ESCON channel-to-channel (CTC)

An ESCON CTC channel is a standard ESCON channel that is defined in HCD as TYPE=CTC. During the initialization of the CHPID, the microcode is loaded into the channel to enable it to support the CTC function on top of the ESCON channel function. There are no external ESCON CTC control units or devices.

As shown in the visual, an ESCON CTC channel must communicate with an ESCON CNC channel—that is, a standard ESCON channel. Note that an ESCON CNC channel communicating with an ESCON CTC channel can also support other ESCON control units, such as DASD through the ESCD.

ESCON CTCs are used to allow inter-processor communications (connects central storage of one MVS image with central storage of another MVS image). This communication can be among different logical partitions in the same CEC, or among different CECs (locally or at long distances).

An ESCON CTC can also be connected point-to-point to an ESCON CNC channel on another processor.

Each ESCON CTC fiber can support up to 512 independent device addresses, each one allocated by a distinct application but sharing the same physical link.

For educational purposes let's imagine an I/O operation running through the CTC path represented in the visual by the CHPID 2A from SYS1 and CHPID 31 from SYS2. An application in SYS1 wants to send data (write) to its peer application in SYS2 which wants to receive such data (read). This operation follows these steps:

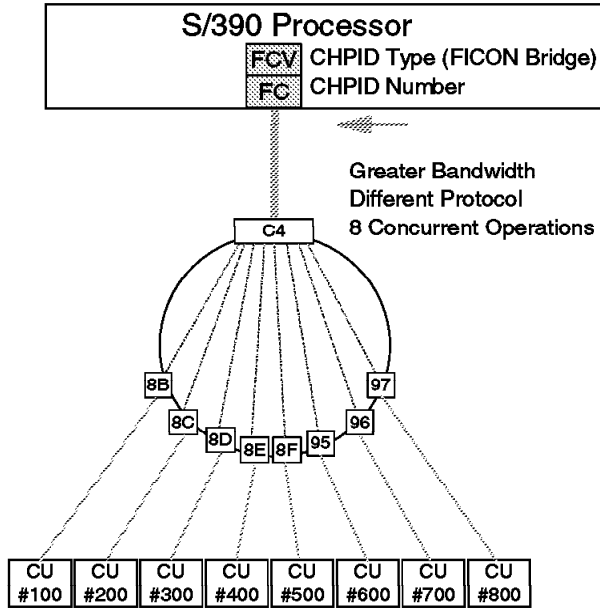
- IOS in SYS1 issues a SSCH in CHPID 2A—the channel program has a Write command. The data transfer is halted.
- IOS in SYS2 receives an I/O interrupt (named attention) in CHPID 31.
- IOS in SYS2 issues a SSCH in CHPID 31—the channel program has a Sense command. The response for that is the information received by IOS in SYS2, that CHPID 2A is issuing a Write command.
- IOS in SYS2 issues a SSCH in CHPID 31—the channel program has a Read command. The data transfers starts.

FICON



★ Two types of FICON channel

★ FICON BRIDGE



★ Native FICON

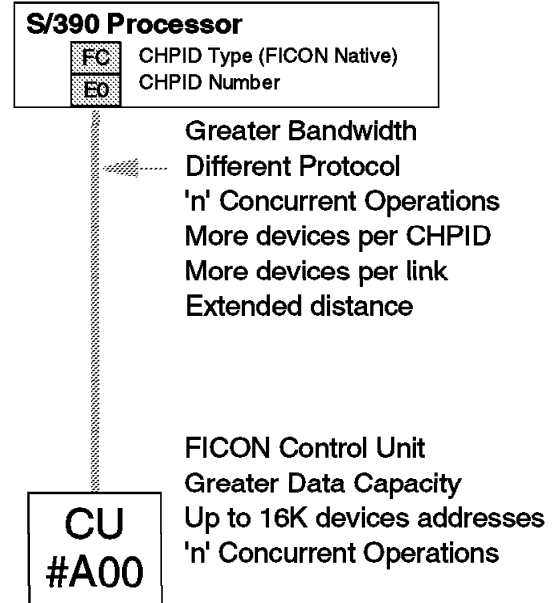


Figure 65. FICON

3.17 Fiber Connection (FICON)

Fiber Connection (FICON) is a new channel protocol introduced in the 9672 G5. This protocol is an ANSI standard that IBM implemented to alleviate:

- ESCON limitations in bandwidth and in the number of devices and control units
- S/390 architecture limit of 256 channels per CEC

FICON supplements, but does not replace ESCON. In a 9672 G6 you may have up to 24 FICON channels. FICON channels use the same fiber optic cables as ESCON channels.

FICON bandwidth is 100 MB/sec full duplex, with 60 to 80 MB/sec expected total data rate. FICON supports more than 4,000 IOs/sec per channel compared to 500 IOs/sec per channel in ESCON. FICON channels have a negligible data rate droop up to at least 43 km, compared to 9 km in ESCON; this means that the FICON data rate does not vary with the distance up to 43 km.

FICON allows a distance (between CEC and control unit) of up to 10 km or 29 km (RPQ) with no repeater to a 9032 ESCD; or distances of up to 100 km with a repeater.

Up to 16,384 devices and up to 256 CUs (per channel) with much fewer connections as less channels, ports, cables, patch panel ports, and so on are required.

At the 9672 G6 announcement time frame, there are no FICON control units yet available; the only type of FICON channel available is the FICON Bridge, also called FCV. The FCV channel is connected to an

ESCD model 5 via a special bridge port card. The bridge converts FICON to ESCON protocol to reach ESCON CUs. One FICON bridge links to eight ESCON switch ports. An FCV channel can multiplex up to eight distinct data transfers from eight ESCON control units. One FCV is equivalent (performance wise) to eight ESCON channels.

In the visual the FICON FCV has a CHPID of FC and connects to bridge port number C4, which connects to eight control units ports.

IBM has a statement of direction for FICON CUs and full FICON ESCDs.

How to Display Channel Types



★ Display CHPID Matrix

```

ISF031I  CONSOLE KHEWITT ACTIVATED
-D M=CHP
IEE174I  10.40.41 DISPLAY M 628
CHANNEL PATH STATUS
  0 1 2 3 4 5 6 7 8 9 A B C D E F
0 . . . . . + + . . + . . .
1 + + + + + + + . . . . .
8 + . . . . + + + + + . . . .
9 + . . + . . . . + + - - + . .
A . . . . + + . . + + + + . . .
B + + + + . . . . + + + + . . .
D . . . . . . . . . + + + +
E + . . . + + + + + + + . . .

***** SYMBOL EXPLANATIONS *****
+ ONLINE      @ PATH NOT VALIDATED  - OFFLINE    . DOES
NOT EXIST
CHANNEL PATH TYPE STATUS
  0 1 2 3 4 5 6 7 8 9 A B C D E F
0 00 00 00 00 00 00 00 0B 0B 00 00 12 00 00 00
1 05 05 09 04 05 05 04 04 12 00 00 00 00 00 00
8 12 00 00 00 05 05 04 04 05 05 05 04 00 00 00
9 05 05 05 04 01 01 01 00 09 05 04 04 12 00 00
A 01 01 01 00 0B 0B 00 00 05 09 04 04 00 00 00
B 05 05 05 04 12 00 00 00 05 05 04 04 01 01 00
D 00 00 00 00 00 00 00 00 12 00 00 00 05 05 04
E 12 00 00 00 05 05 05 04 09 05 04 04 01 01 00

***** SYMBOL EXPLANATIONS*****
00 UNKNOWN
01 PARALLEL BLOCK MULTIPLEX
02 PARALLEL BYTE MULTIPLEX
03 ESCON POINT TO POINT
04 ESCON SWITCH OR POINT TO POINT
05 ESCON SWITCHED POINT TO POINT
06 ESCON PATH TO A BLOCK CONVERTER
07 NATIVE INTERFACE
08 CTC POINT TO POINT
09 CTC SWITCHED POINT TO POINT
0A CTC SWITCHED OR POINT TO POINT
0B COUPLING FACILITY SENDER
0C COUPLING FACILITY RECEIVER
0D UNKNOWN
0E UNKNOWN
0F ESCON PATH TO A BYTE CONVERTER
10 RESERVED
11 RESERVED
12 OPEN SYSTEMS ADAPTER
13 INTERNAL SYSTEM DEVICE
14 RESERVED
15 RESERVED
16 CLUSTER BUS SENDER
17 CLUSTER BUS RECEIVER
18 INTERNAL COUPLING SENDER
19 INTERNAL COUPLING RECEIVER
NA INFORMATION NOT AVAILABLE
  
```

★ Channel type Chpid 9C = OSA

Figure 66. How to display channel types

3.18 How to display channel types

The DISPLAY MATRIX CHPID operator MVS command provides information about the status and type of channels. There are two parts to the display:

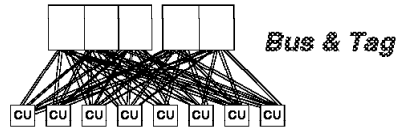
1. The first section displays the channel path status. The channel path status is relative to the system where the command is issued. That is, a CHPID may be displayed as being offline, but if this CHPID is shared, using EMIF, by other logical partitions, it may not be offline physically.
2. The second section displays the channel path type. Note that where the first section only displays the status of channels available to the MVS image, the second section provides information about all channels installed on the processor.

S/390 Connectivity



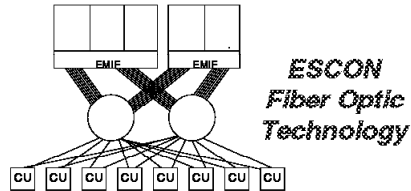
★ Parallel Channels

- ▶ 1.5MB/sec --> 4.5MB/sec
- ▶ 400 Feet/122 Meters Distance Limitation
- ▶ Disruptive Change Management
- ▶ Limited Connectivity



★ ESCON Channels

- ▶ 20MB/sec Link Rate
 - 10MB/sec --> 17MB/sec Data Transfer Rate
 - Half duplex
- ▶ 43 KM/26.7 Mile Maximum Distance
 - Performance Degradation at 9 KM/5.5 Miles
- ▶ Switched Point to Point Topology - Connectivity
- ▶ ESCON Multiple Image Facility (EMIF)
- ▶ System Automation for OS/390



★ FICON Channels

- ▶ Strengths of ESCON Plus More!
- ▶ 100MB/sec Link Rate
 - Full Duplex
- ▶ Multiple Concurrent I/Os on One Channel
- ▶ 43 KM/26.7 Mile Maximum Distance
 - Negligible Performance Degradation
- ▶ Coexists and Exploits Existing ESCON Infrastructure
- ▶ Positions for S/390 CMOS MIP and Data Growth

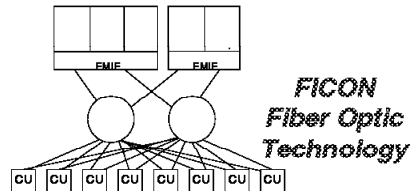


Figure 67. S/390 connectivity

3.19 S/390 connectivity

This visual summarizes the three different types of channel technology available on S/390 systems.

System-Related I/O Work Flow

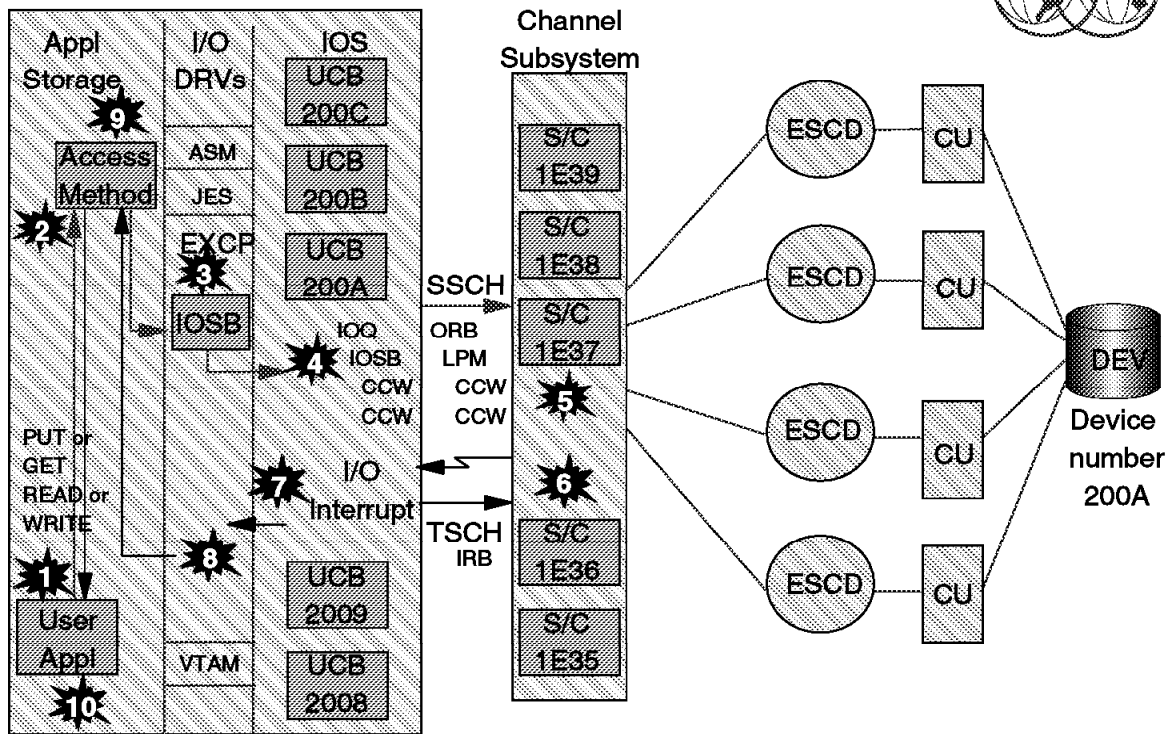


Figure 68. System-related I/O work flow

3.20 System-related I/O work flow

This visual shows the flow of an I/O operation from the request done by an application until the operation completes.

1. The user program accesses a data set by issuing an OPEN macro instruction. To request either input or output of data it uses an I/O macro instruction like GET, PUT, READ, or WRITE, and specifies a target I/O device. An I/O macro instruction invokes an access method. An access method has the following functions:
 - Writes the channel program (with virtual addresses)
 - Implements buffering
 - Guarantees synchronization
 - Executes I/O recovery

The user program could bypass the access method, but it would then need to consider many details of the I/O operation, such as the physical characteristics of the device.

2. There are several MVS access methods, each of which offers differing functions to the user program. The selection of an access method depends on how the program plans to access the data (randomly, or sequentially, for example)
3. To request the movement of data, either the access method or the user program presents information about the operation to an I/O driver routine (usually the EXCP driver) by issuing the EXCP macro instruction. The I/O driver translates the channel program from virtual to real (a

format acceptable by the channel subsystem) fixes the pages containing the CCWs and the data buffers, guarantees the volume extents, and invokes the I/O Supervisor (IOS).

4. IOS, if there is not a pending I/O operation for the required device (originated previously in this system), issues the Start Subchannel (SSCH) instruction to the channel subsystem. Then, the CPU continues with other work (the task executing the access method is probably placed in wait state) until the channel subsystem indicates with an I/O interrupt that the I/O operation has completed. If the device is busy the request is queued in the UCB control block.
5. The SAP selects a channel path to initiate the I/O operation. This channel executes the channel program controlling the movement of data between device, control unit, and central storage.
6. When the I/O operation is complete, SAP signals the completion by generating an I/O interrupt.
7. IOS processes the interruption by determining the status of the I/O operation (successful or otherwise) from the channel subsystem. IOS indicates that I/O is complete by posting the waiting task and calling the dispatcher.
8. When appropriate, the dispatcher dispatches the task returning to the access method code.
9. The access method returns control to the user program, which can then continue its processing.

What is HCD



- ★ Combines hardware (IOCP) and software (MVSCP) definitions into a single process
- ★ Allows multiple software and processor configurations to be defined in a single database
- ★ Creates a single definition for each physical control unit and device
- ★ Provides immediate online validation of configuration data
- ★ Contains extensive online help
- ★ Provides dynamic I/O reconfiguration

Figure 69. What is HCD

3.21 What is HCD

The channel subsystem (CSS) and the IBM OS/390 operating system need to know what hardware resources are available in the computer system and how these resources are connected. This information is called hardware configuration.

Before HCD was available, you had to use IOCP to define the hardware configuration to the channel subsystem and the MVS Configuration Program (MVSCP) to define the hardware configuration to the MVS operating system (also called software definition).

The configuration you define with HCD may consist of multiple processors, each containing multiple partitions. HCD stores the entire configuration data in a central repository, the input/output definition file (IODF). The IODF as single source for all hardware and software definitions for a multi-processor or multi-partition system eliminates the need to maintain several independent MVSCP or IOCP data sets. That means, that you enter the information only once using an interactive dialog.

HCD is part of OS/390. It needs a running OS/390 system before it can be used to define a hardware configuration. Therefore, an installation should first load an OS/390 system, using an old IODF, or a ServerPac Starter IODF to IPL the OS/390 system for the first time. Existing MVSCP configuration data has to be migrated into an IODF and is then used to IPL the system. HCD can be used on that system to define the full configuration.

Hardware Configuration Definition (HCD) provides an interactive interface that allows you to define the hardware configuration to both the channel subsystem and the operating system.

HCD combined with the Dynamic I/O Reconfiguration Management function allows you to make changes to either the hardware or software configurations without the need of a power-on reset (POR) or an IPL. This greatly enhances overall system availability by making possible the elimination of scheduled outages that were previously necessary when parts of the configuration were changed.

Note: This section on HCD illustrates the many panels used in creating a new I/O configuration. It is not a complete definition and does not show all of the panels and the various options needed. It is an attempt to make you familiar with the concepts available with HCD.

See *OS/390 Hardware Configuration Definition: User's Guide*, SC28-1848, and *OS/390 Hardware Configuration Definition Planning*, GC28-1750, for how to use HCD.

HCD Functions

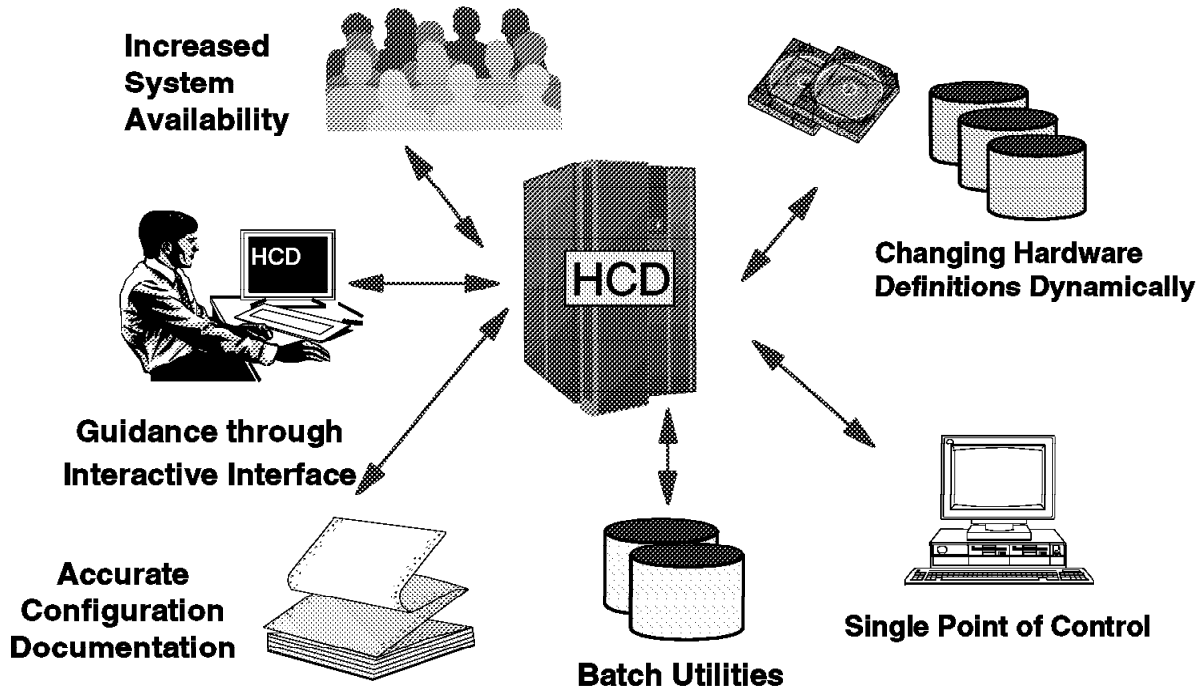


Figure 70. HCD functions

3.21.1 HCD functions

Single point of control: With HCD you have a single source, the IODF production data set, for your configuration data. This means that hardware and software definitions as well as ESCON director definitions can be done from HCD and can be activated with the data stored in the IODF. However, to change the ESCD matrix switch configuration, HCD needs the ESCON Manager product.

Increased system availability: HCD checks the configuration data when it is entered and therefore reduces the chance of unplanned system outages due to inconsistent definitions.

Changing hardware definitions dynamically: HCD offers dynamic I/O reconfiguration management. This function allows you to change your hardware and software definitions on the fly—you can add devices, or change devices, channel paths, and control units, without performing a POR or an IPL. You may also perform software-only changes, even if the associated hardware is not installed.

Sysplex-wide activate: HCD offers you a single point of control for systems in a sysplex. You can dynamically activate the hardware and software configuration changes for systems defined in a sysplex, if the CECs are in the same HMCplex (also called an S/390 Microprocessor Cluster).

Accurate configuration documentation: The actual configuration definitions for one or more processors in the IODF are the basis for the reports you can produce with HCD. This means that the reports are accurate and reflect the up-to-date definition of your configuration. HCD provides a number of textual reports and graphical reports, that can be either printed or displayed. The printed output can be used

for documentation purposes providing the base for further configuration planning tasks. The display function allows you to get a quick overview of your logical hardware configuration.

Guidance through interactive interface: HCD provides an interactive user interface, based on ISPF, that supports both the hardware and the software configuration definition functions. The primary way of defining the configuration is through the ISPF dialog. HCD consists of a series of panels that guide you through all aspects of the configuration task. The configuration data is presented in lists. HCD offers extensive online help and prompting facilities. Help includes information about panels, commands, data displayed, available actions, and context-sensitive help for input fields. A fast path for experienced users is also supported.

Batch utilities: In addition to the interactive interface, HCD also offers a number of batch utilities. You can use these utilities, for instance, to migrate your existing configuration data; to maintain the IODF; or to print configuration reports.

Cross operating system support: HCD allows you to define both OS/390 and VM/ESA configurations from OS/390.

Support for controlling CECS in HMCplex: HCD provides functions for IOCDS and IPL attributes management, which simplify the configuration and operational support for those CECs which service elements that are in the same HMC LAN. IOCDS are data sets allocated in the service DASD from the Service Element in a 9672. The I/O configuration is copied from an IOCDS at Power-on Reset time to HSA.

Dynamic I/O Reconfiguration



- ★ Static devices
- ★ Dynamic devices
- ★ Installation-static devices
- ★ Hardware System Area (HSA)
- ★ Configuration token
- ★ Software-only changes
- ★ Software and hardware changes

Figure 71. Dynamic I/O reconfiguration

3.21.2 Dynamic I/O reconfiguration

Dynamic I/O reconfiguration is the ability to select a new I/O configuration definition without having to perform a power-on reset (POR, the load of a processor's microcode plus the load of the I/O configuration) of the hardware or an IPL of the OS/390 system. It allows an installation to add, delete, or modify the definitions of channel paths, control units, and I/O devices to the software and hardware configurations. You can change the I/O configuration definitions for both software and hardware or for software only. It also allows you to dynamically change:

- Eligible device table (EDT), where you associate a set of devices with an esoteric name.
- Device preference List, where you define a sequenced priority of device types, to be used if you are allocating a new data set and you have more than one possible device type as candidate.

To use dynamic I/O reconfiguration, HCD must be used to define the configuration.

To implement a dynamic I/O reconfiguration you need to do the following steps:

1. Define an IODF production with the full new configuration, including the devices that will not be changed, the new ones, and the ones to be changed. The deleted devices are omitted.
2. Issue the ACTIVATE command (through the MVS console or HCD). MVS reads the new configuration and generates a delta list, that includes the new elements and the ones to be deleted. MVS synchronously updates the software configuration and the hardware configuration (using a special instruction able to update the HSA).

Dynamic I/O reconfiguration has the following benefits:

- It increases system availability by allowing you to change the I/O configuration while OS/390 is running, thus eliminating the POR and IPL for selecting a new or updated I/O configuration.
- It allows you to make I/O configuration changes when your installation needs them, rather than waiting for a scheduled outage to make changes.
- It minimizes the need to logically define hardware devices that do not physically exist in a configuration (also known as over-defining a configuration).

Dynamic I/O Device Types



★ Static

- ▶ UIM does not support Dynamic I/O changes

★ Installation-static

- ▶ UIM supports Dynamic I/O
 - Specified with DYNAMIC=NO in HCD

★ Dynamic

- ▶ UIM supports Dynamic I/O
 - Specify DYNAMIC=YES in HCD

Figure 72. Dynamic I/O device types

3.21.3 Dynamic reconfiguration concepts

This section introduces and explains a number of dynamic reconfiguration management concepts. These concepts and terms are used in subsequent sections of this chapter when discussing dynamic I/O reconfiguration:

- *Unit Information Module (UIM)*

IOS does not understand the peculiarities of each possible device type. Each device type that exists has an associated software routine called the unit information module (UIM), which is included in the IOS code. UIM contains the device support code. UIMs are invoked by IOS at device initialization and by HCD for testing a specific I/O configuration that you may have. One of the functions of UIM is to specify whether or not the device type supports dynamic I/O configuration.

- *Device types*

OS/390 treats device control information (for example UCBs) differently depending upon whether the device can be dynamically reconfigured or not. This state may be also declared by you through the DYNAMIC parameter at IODEVICE in IOCP. The possible device types are:

Static

A static device is a device whose type as defined in its UIM does not support dynamic I/O reconfiguration. A static device cannot be dynamically added, deleted, or modified in the software configuration definition. Therefore, the device is not available for use until the next IPL of MVS. However, the device can be added, deleted, and modified in the hardware configuration definition, but its channel path information cannot be modified.

Installation-static An installation-static device is a device whose type as specified in its UIM supports dynamic I/O configuration, but which you have defined as DYNAMIC=NO (default is NO) in the HCD definition. You might specify DYNAMIC=NO if your installation has programs, including customer programs, supplier programs and products, that depend on device-related data structures such as UCB and EDT, or use existing old MVS services which access these data structures, and are unprepared to handle dynamic changes to these structures.

Installation-static devices can be dynamically added to the software I/O configuration, but can not be deleted or modified while OS/390 is running. Devices can be dynamically added, deleted, and modified in the hardware configuration definition, but their channel path information cannot be modified. The control information for installation-static devices can be accessed using either old UCB services or the new ones. Defining devices as installation-static should be considered as an interim measure until all the programs accessing their UCBs have been converted to the new UCB services.

Note: Modifying or deleting an installation-static device requires two separate dynamic I/O configuration changes:

1. Change the installation-static device to dynamic (by activating a new IODF that defines the device as dynamic). No other device characteristics for that device can be changed on the redefinition activation.
2. Delete or modify the dynamic device (by activating another IODF that contains the appropriate changes to the device). If the device is being modified, and you want it to be changed back to installation-static after the change, you can do this at the same time that other changes are being activated by specifying DYNAMIC=NO as part of the change.

Dynamic A dynamic device is a device whose device type as defined in its UIM supports dynamic I/O configuration (in software and hardware). In order to dynamically add, modify, or delete such a device while OS/390 is running, it is also necessary that the device be defined as dynamic in HCD (DYNAMIC=YES).

- *Hardware System Area (HSA)*

When you perform a POR, the hardware I/O configuration definition is loaded from an IOCDs into the HSA. HSA is a non-software addressable piece of CS, used for a microcode work-area, LPAR processing, and I/O configuration data. When you activate a new configuration, HSA is updated to reflect the new hardware I/O configuration definition.

If you plan to use dynamic I/O reconfiguration to add equipment to your configuration, you must specify an expansion factor for the HSA size before initiating the POR. The expansion factor defines how much additional HSA storage is reserved for future dynamic I/O configuration changes.

- *Eligible device table (EDT)*

During dynamic I/O reconfiguration it is possible to also change the EDT. Usually, OS/390 uses one EDT to process allocation requests. However, when you dynamically change your system's software I/O configuration definition, OS/390 may have to use two EDTs to process the change:

- The primary EDT processes all current and new allocation requests. OS/390 runs with only the primary EDT until you make a dynamic I/O configuration change. OS/390 activates a new primary EDT for the new I/O configuration, which makes the former primary EDT the secondary EDT.
- The secondary EDT receives no new allocation requests. The system deletes the secondary EDT when it finishes the allocation requests issued before the dynamic I/O change.

IODF



★ IODF File

▶ Naming conventions

- SYSn.IODFnn.WORK for work IODF
- SYSn.IODFnn for production IODF

★ IODF File is a VSAM data set

▶ catalog considerations

★ Backup

★ Placement

▶ Non SMS managed volume recommended

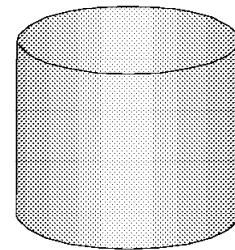


Figure 73. IODF file

3.22 IODF data set

The I/O definition file (IODF) is a VSAM linear data set that is built and maintained by HCD. There are two types of IODF data set:

- *Work IODF*

A work IODF allows you to create a new I/O configuration definition or modify an existing one. You can have multiple work IODFs, each with a unique name.

All work IODF names must conform to the following convention:

```
' hhhhhhhh.IODFxx.yyyyyyyy.yyyyyyyy'
```

Where:

hhhhhhhh High level qualifier—up to eight characters

xx Hexadecimal number in the range of 00 through FF

yyyyyyyy Optional qualifiers—up to eight characters each

HCD provides considerable flexibility in the choice of work IODF names. It is, however, recommended that you be more stringent in your naming conventions. It is recommended that you use the following convention when choosing a name for an IODF:

```
' hhhhhhhh.IODFxx.WORK'
```

In this manner, you can easily ascertain the nature of the IODF.

- *Production IODF*

A production IODF is the data set used to obtain the definitions needed to run the system. A production IODF is a read-only data set, which preserves the integrity of the IODF.

The naming convention of the production IODF is more restrictive. If the IODF is to be used for IPL and dynamic activation, the production IODF must have the following format:

`' hhhhhhhh.IODFxx'`

Where:

hhhhhhhh The high-level qualifier

xx The IODF suffix.

The production IODF is an IPLable image of the work IODF. The device number of the device where the IODF is located is specified in the *load parm* at the HMC console. In the LOADxx parmlib member you specify:

xx: The suffix

hhhhhhhh The high level qualifier

The following naming convention is a good way to relate the current production IODF to its work IODF, and is easily extendable as new IODF versions are created:

| | |
|--------------------------------|--------------------------------|
| ACTIVE PRODUCTION IODF: | <u>' hhhhhhhh.IODF30'</u> |
| SUBSEQUENT WORK IODF: | <u>' hhhhhhhh.IODF31.WORK'</u> |
| RESULTING NEW PRODUCTION IODF: | <u>' hhhhhhhh.IODF31'</u> |

3.22.1 Catalog considerations

In a multisystem environment, a single IODF can define several processor software configurations. Because an IODF is a VSAM data set, it can be cataloged in only one catalog. Therefore, if you wish to share an IODF data set among multiple systems and each system is using a separate master catalog, you must define (in the master catalog of each system) an alias that relates to the user catalog on DASD that is shared among the systems. Define aliases and the user catalog before using HCD to define IODF data sets.

3.22.2 Production IODF backup

It is suggested that you maintain a backup copy of your production IODF on a separate volume that is accessible to all systems that are sharing the backup. When the primary IODF volume is inaccessible or the IODF itself is corrupted, the system can be IPLed through a backup IODF on the alternate IODF volume.

3.22.3 IODF placement

The production IODF to be used during IPL processing must be placed on the IODF device. The IODF device is pointed to by the load parm (L2) value on the IPL window of your hardware management console.

In an SMS environment, care should be taken to ensure that either:

- The production IODF data set name is not managed by SMS. You can then specify the IODF volume serial number when creating a production IODF.
- The ACS routines are set up to automatically place the production IODFs on the IODF volume.

3.22.4 Multiple configurations in a single IODF

The installation must decide whether to create one IODF for each CEC or to combine the I/O definitions for two or more CECs in a single IODF. The advantage of including two or more CECs with shared devices in one IODF is that the information for shared devices has to be entered only once.

A second reason to consider maintaining I/O configuration definitions for several CECs in one IODF is that it allows you to easily move I/O configurations from one partition to another (in either the same CEC or separate CECs).

The general recommendation is to create one IODF for a *logical processor complex*; that is a group of processors that share I/O devices, as a sysplex. In this way, one MVS system can be used for updating the IODF with the configuration changes, which minimizes the chance of error.

If you previously have had multiple IODFs defining your complex, the HCD copy/merge function can be used to copy relevant parts of configuration data from a source to a target IODF, or even from areas within an IODF.

Definition Order



- ★ Recommended definition order within HCD
 - ▶ Operating system
 - EDTs
 - Esoterics
 - ▶ Switches
 - Ports
 - ▶ Processor
 - Partitions (LPAR only)
 - Channels (chpids)
 - ▶ Control units
 - ▶ Devices
 - ▶ Consoles

Figure 74. Definition order

3.22.5 Definition order

You can define the objects of a configuration in almost any order but at one point you have to connect objects together. You can only connect objects that are already defined; therefore, it is useful to define the objects in a logical order. For example, when defining I/O devices during the hardware definition, you are prompted to add devices to existing operating system definitions. Therefore, it is useful to define the operating system before the devices.

The suggested sequence to define a configuration is shown in this visual.

HCD Primary Option Menu



```
OS/390 Release 5 HCD
Command ===>
-----
Hardware Configuration

Select one of the following.

1  1. Define, modify, or view configuration data
   2. Activate or process configuration data
   3. Print or compare configuration data
   4. Create or view graphical configuration report
   5. Migrate configuration data
   6. Maintain I/O definition files
   7. Query supported hardware and installed UIMs
   8. Getting started with this dialog
   9. What's new in this release

For options 1 to 5, specify the name of the IODF to be used.

I/O definition file . . . 'SYS6.IODF27.WORK' +
```

Figure 75. HCD primary option menu

3.23 HCD primary option menu

HCD provides an action bar-driven interface that exploits many of the usability features of the Common User Access (CUA) interface.

To select an item from a numbered selection list, type the number you want to select in the input field (left of the first list item) and press the Enter key. An example of a numbered list is the HCD primary task selection panel, shown in the visual. This panel is displayed when you start an HCD session.

To create a new IODF, specify as follows:

```
I/O definition file . . . 'SYS6.IODF27.WORK'
```

Then, enter 1 in the input field and press the Enter key.

On entering HCD, you are presented with the HCD primary menu panel. The first time you use HCD, you must enter the IODF data set name that you wish to use. If this is not the name of an existing IODF, HCD creates a new work IODF for you.

If you have a production IODF, it is advisable to start from the active production IODF as a base. This is mandatory for a dynamic environment.

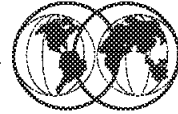
After entering the new IODF name enclosed in quotes, for example:

```
'SYS6.IODF27.WORK'
```

HCD then displays the Create Work IODF Definition File panel.

The following visuals will step you through an IODF creation.

Creating a New Work IODF



```
OS/390 Release 5 HCD
Create Work I/O Definition File

CBDPUT10

The specified I/O definition file does not exist. To create a new
file, specify the following values.

IODF name . . . . . 'SYS6.IODF27.WORK'

Volume serial number . TOTNEW +

Space allocation . . . 1024 (Number of 4K blocks)

Activity logging . . . Yes (Yes or No)

Description . . . . . New IODF for MVSNEW_____
_____
_____

F1=Help   F2=Split  F3=Exit   F4=Prompt  F9=Swap   F12=Cancel
```

Figure 76. How to create a new work IODF

3.23.1 How to create a new work IODF

Specify the volume you wish the IODF to be allocated to (prompting is available for this field) and press F4 for a list of volumes. If you are working within an SMS-managed environment, the volume serial number is not required.

Specify whether you want activity logging to be enabled. If you opt for this function, then the activity log will be displayed as the last panel when exiting following an update to the IODF.

At this time you must also specify the size of the new work IODF in terms of the number of 4 KB blocks.

Defining Configuration Data

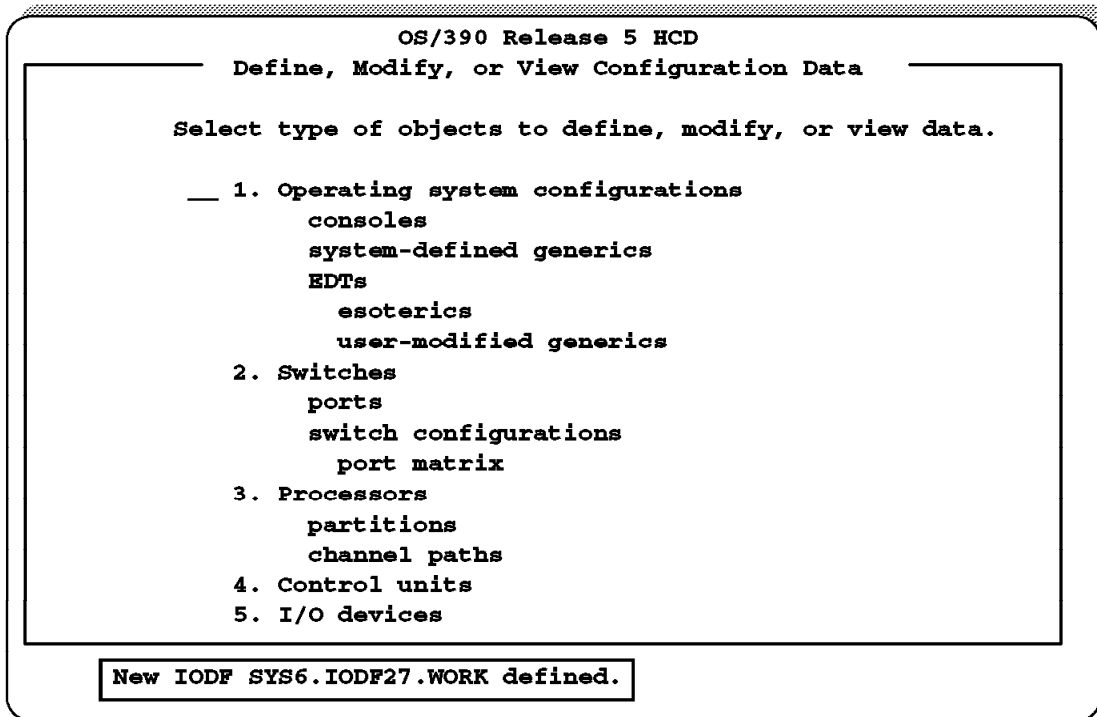


Figure 77. Defining configuration data

3.23.2 Defining configuration data

The next panel displayed after choosing Option 1 from the primary option menu is shown in the visual. This panel displays all the objects for which the HCD dialog provides action lists where you can define the characteristics and the relation between the objects.

From this panel, you can go step-by-step to define, change, prime, or delete the following:

- Operating system configurations
- EDTs
- Generics
- Esoteric groups
- Processors
- Partitions (LPARs)
- Channel paths
- Control units
- Devices
- Consoles

Before using the dialog of HCD to define a hardware configuration, you should have a plan of what your configuration should look like, and what you have to do to accomplish that. Preferably, the requirements of your configuration should be established in a configuration plan. Refer to *OS/390 Hardware Configuration Definition Planning*, GC28-1750, for an OS/390 or MVS configuration.

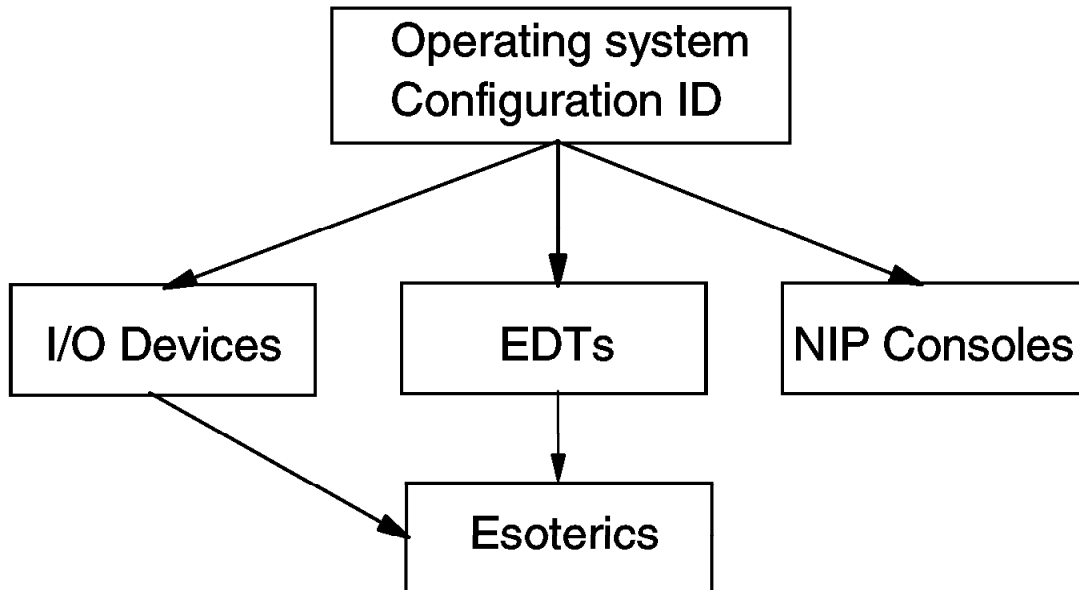


Figure 78. Operating system definition

3.24 Operating system definition

It is recommended to define the operating system configuration before you define anything else. An operating system (OS) configuration defines the data that is used by MVS, to build its IO control blocks. An IODF can contain more than one OS configuration. At IPL time one of them is selected.

This visual shows the OS elements (I/O devices, EDTs, Esoteric, and NIP consoles) and their relationship within the MVS operating system. A NIP console is the console used by the nucleus initialization program (NIP), an MVS code running at IPL time. NIP is in charge of initializing all MVS components and subsystems. At early stages of this initialization there are no MCS consoles available for communication with the operator. The NIP console is then used for such communication. The message Specify System Parameter is displayed on this device.

At IPL the LOADxx parmlib member identifies the operating system configuration (you may have more than one per IODF) and the EDT identifier.

How to Define a New Config ID

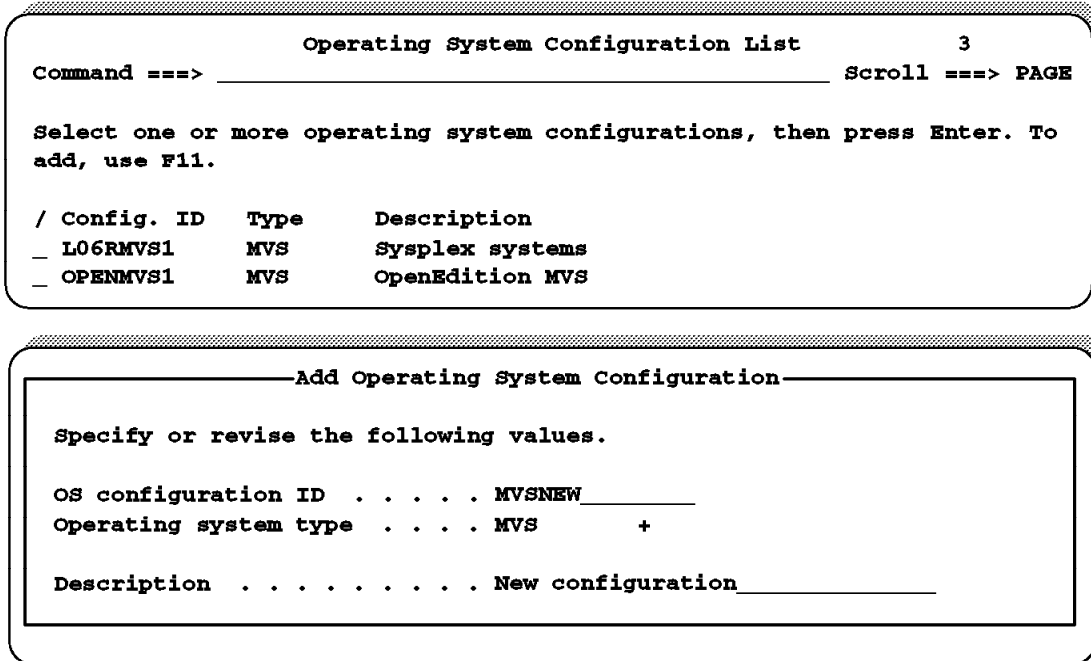


Figure 79. How to define an operating system

3.24.1 How to define an operating system

From the HCD primary menu, shown in Figure 75 on page 117 select:

- 1 Define, modify, or view configuration data

When the next panel appears, then select:

- 1 Operating system configurations.

The visual shows two configurations already defined. To add a new configuration, use F11.

3.24.1.1 Add Operating System Configuration panel

Complete the Add Operating System Configuration panel with the name of the operating system you would like to define. In the visual, we have defined **MVSNEW**.

Having defined the operating system you can now define the EDTs.

EDT and Esoteric

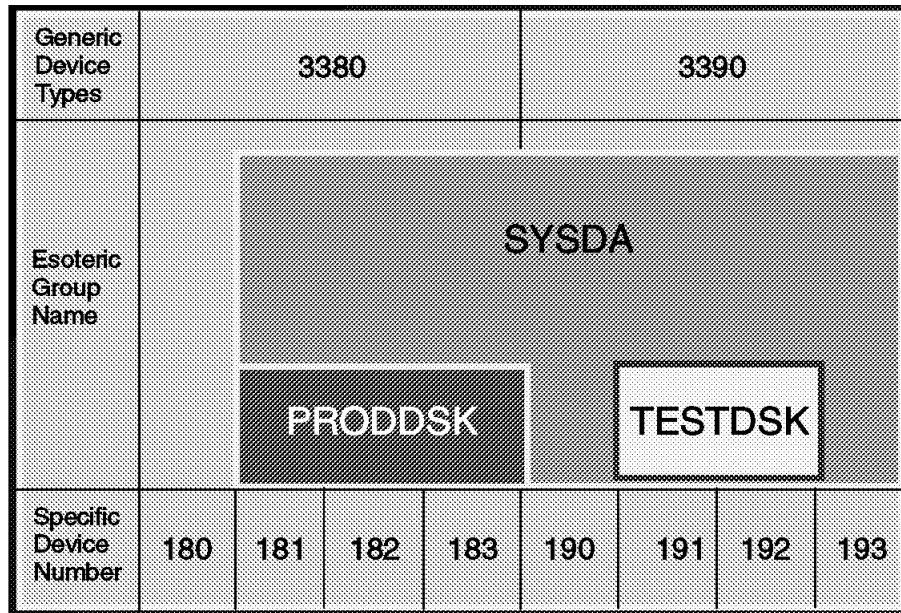


Figure 80. EDT and esoteric

3.24.2 EDT and esoterics

An EDT is a list of devices with esoteric names. It is used at allocation time to generate a list of the devices that are candidates to receive the data set. An MVS operating system may have more than one EDT but only one can be active at any one time. In the visual, the EDT would have an esoteric list of SYSDA, PRODDSK, and TESTDSK.

3.24.2.1 Understanding I/O device allocation in MVS

When you submit a job, you identify I/O devices required by the job. The device information can be obtained from either a catalog (if the data set already exists), SMS overrides, or specific UNIT parameters on DD statements (for new data sets). Before the job can continue execution, an MVS initiator must allocate all those devices to the job.

There are three ways to specify device allocation for a job using the UNIT parameter on a DD card.

- An specific device number
- A generic device type
- An esoteric device group

3.24.2.2 Indicating a specific device for allocation

To request a device explicitly for a job, specify a device number on the UNIT= parameter or on the corresponding dynamic allocation parameter. If that device is available, MVS allocates the device to the job. However, if the device is unavailable (for example, a tape drive allocated to another job), your job must wait until the device is available for allocation.

3.24.2.3 Specifying a generic device type for allocation

MVS logically groups device types with similar characteristics and assigns the group a generic name. Such a group is called a generic device type. MVS, for example, groups the 3390-1 and 3390-2 into a generic device type named 3390. Any time a program allocates a 3390, MVS interprets it to mean any of the devices in that generic device type.

To request a device allocation, you can specify a generic device type on the UNIT= parameter. MVS allocates a device from the specified generic device type. For example, if you code the following DD statement:

```
//OUTPUT DD UNIT=3390,...
```

MVS allocates a device from generic device type 3390. Generic device type 3390 should not be confused with specific device number 3390. To avoid having your specification misinterpreted as a specific device number use the following notation for the device number:

```
//OUTPUT DD UNIT=/3390,...
```

3.24.2.4 Specifying an esoteric device group for allocation

A job that specifies an esoteric device group is requesting MVS to allocate a device from that group. An esoteric device group can include devices of different generic device types.

The *device preference list* indicates the preferred sequence of device types candidates to allocate the data set. It is used when the esoteric definition has more than one device type.

All of the devices that you assign to an esoteric device group must be of the same device class with the following exception: you can define esoteric device groups that contain devices from both DASD and tape device classes.

Devices belong to one of the following classes:

- Channel-to-channel adapters
- Communication devices
- Direct access devices
- Display devices
- Character readers
- Tape devices
- Unit record devices

To request device allocation, you can specify an esoteric device group on the UNIT= parameter on the DD JCL statement.

In this visual SYSDA is the esoteric group name for three 3380s (device numbers 181, 182, and 183) and four 3390s (device numbers 190 through 193). When UNIT=SYSDA appears on a DD statement, units 181, 182, 183, 190, 191, 192, and 193 are eligible candidate devices.

TESTDSK is the esoteric group name for two 3390 DASDs (device numbers 191 and 192).

PRODDSK is the esoteric group name for three 3380 DASDs (device numbers 181, 182, and 183).

Select New Configuration for EDTs



```
Operating System Configuration List          3
Command ==> _____ Scroll ==> PAGE

Select one or more operating system configurations, then press Enter. To
add, use F11.

/ Config. ID  Type  Description
_ L06RMVS1    MVS   Sysplex systems
_ OPENMVS1    MVS   OpenEdition MVS
/ MVSNEW      MVS   New configuration
```

Figure 81. Select new configuration for EDTs

3.24.3 Select new configuration for EDTs

Before you can define EDTs, you must have defined an operating system. You define an EDT as follows:

- On the primary task selection panel, select Define, modify, or view configuration data and on the resulting panel the object Operating system configurations. HCD displays the operating system configuration list of all operating system configurations currently defined in the IODF.
- On the Operating System Configuration List panel, select the OS configuration, shown in the visual as **MVSNEW**. The next panel displayed allows you to select to work with EDTs, as shown in Figure 82 on page 126.

How to Define an EDT (1)



```
Goto Backup Query Help
----- Actions on selected operating systems

Command ==>
Select one or
add, use F11.
/ Config. ID
_ L06RMVS1
_ OPENMVS1
/ MVSNEW
*****

_5_ 1. Add like . . . . . (a)
    2. Repeat (copy) OS configurations . . (r)
    3. Change . . . . . (c)
    4. Delete . . . . . (d)
    5. Work with EDTs . . . . . (s)
    6. Work with consoles . . . . . (n)
    7. Work with attached devices . . . . (u)
    8. View generics by name . . . . . (g)
    9. View generics by preference value . (p)

F1=Help    F2=Split    F3=Exit    F9=Swap    F12=Cancel
```

Figure 82. How to define an EDT

3.24.4 How to define an EDT

You define an EDT as follows:

- On the primary task selection panel, select Define, modify, or view configuration data and on the resulting panel the object Operating system configurations. HCD displays the operating system configuration list of all operating system configurations currently defined in the IODF.
- On the Operating System Configuration List panel, select the OS configuration and the Work with EDTs action from the context menu (or action code s). HCD displays the EDT List panel.
- Select an operating system from the OS Configuration List panel, using the **F** action key and then select **Work with EDTs** Option 5 from the context menu. This displays the panel shown in Figure 84 on page 128.

How to Define an EDT



```

                                EDT List
      Goto  Backup  Query  Help
-----
                                Row 1 of 1
Command ===> _____ Scroll ===> PAGE

Select one or more EDTs, then press Enter. To add, use F11.

Configuration ID . : MVSNEW      New configuration
/ EDT Last Update By      Description

```

Figure 83. How to define an EDT

3.24.5 How to define an EDT

For an MVS operating system, you have to define at least one eligible device table (EDT). An EDT can consist of one or more esoteric device groups and names of the generic device types. Esoteric device groups are installation-defined groupings of I/O devices.

An OS configuration can contain more than one EDT; OS/390 is told which one to use at IPL time. For background information about I/O device allocation in MVS that you need to understand before defining EDTs and esoteric groups, refer to *OS/390 Hardware Configuration Definition Planning*, GC28-1750.

Define EDT Identifier



```
EDT List
Goto Backup Query Help
Add EDT
CBDPED10
Specify the following values.
Configuration ID . : MVSNEW      New configuration
EDT identifier . . . 0A
Description . . . . New EDT for MVSNEW

-----
EDT List
Goto Backup Query Help
-----
Row 1 of 1
Command ==> _____ Scroll ==> PAGE
Select one or more EDTs, then press Enter. To add, use F11.
Configuration ID . : MVSNEW      New configuration
/ EDT Last Update By      Description
  OA 00-02-25 ROGERS      New EDT for MVSNEW
```

Figure 84. Define EDT identifier

3.24.6 Define EDT identifier

- On the EDT LIST panel, press F11 to add an EDT.
- Enter the EDT identifier (any two alphanumeric characters) and the description for the EDT.
- Pressing Enter redisplay the EDT List panel with the just defined EDT.

How to Add an Esoteric

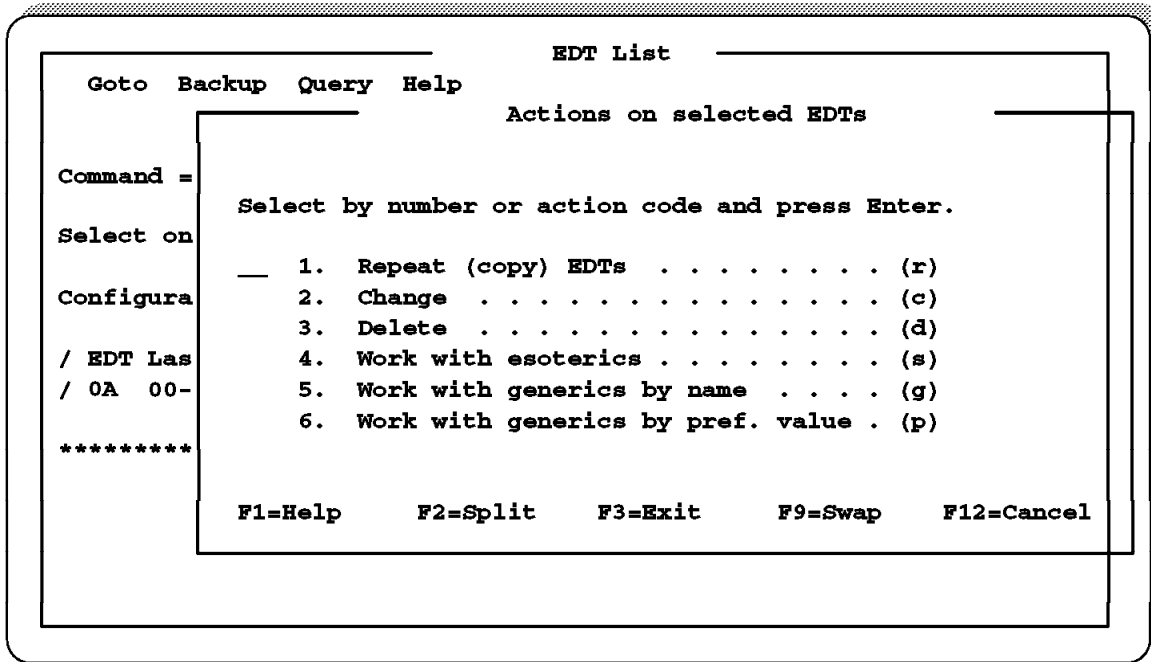


Figure 85. How to add an esoteric

3.24.7 How to add an esoteric

An esoteric device group identifies the I/O devices that are included in that group. The name you assign to an esoteric device group is called the esoteric name. To request allocation of a device from an esoteric device group, specify the esoteric name on the UNIT parameter of a JCL DD statement. The name esoteric device group is often shortened to esoteric group or simply esoteric.

To add an esoteric, select Option 5 **Work with EDTs** shown in Figure 82 on page 126. Then select the EDT identifier using the **F7** action key. Then select **Work with Esoterics** from the context menu. Continue with Figure 86 on page 130.

Add an Esoteric



```

                                     Esoteric List
-----
Goto  Filter  Backup  Query  Help
-----
Command ===> _____ Scroll ===> PAGE

Select one or more esoterics, then press Enter. To add, use F11.

Configuration ID . . : MVSNEW      Sysplex systems
EDT identifier . . . : 0A

/ Esoteric  VIO    Token  State
***** Bottom of data *****

```

```

                                     Add Esoteric
-----
Specify the following values.

Esoteric name . . . . SYSDA____
VIO eligible . . . . No      (Yes or No)
Token . . . . . . . . _____

```

Figure 86. Add an esoteric

3.24.8 Add an esoteric

The Esoteric List panel appears. Press F11 to add an esoteric. The visual shows **SYSDA**.

3.24.8.1 Esoteric token

The esoteric token is an optional value. In the past there have been access problems with data sets cataloged with an esoteric device group name. HCD arranges esoterics alphabetically, but the catalog contains the EDT index entry pointing to the esoteric. After HCD has reordered the esoterics, allocation searches the incorrect device for a data set. If you specify an esoteric token, this token will be used as the entry point to the catalog. Specify the token such that your existing esoteric or non-alphabetic order is maintained.

3.24.8.2 Virtual I/O (VIO)

VIO refers to data set allocations that exist in paging storage only. MVS does not use a real device unless MVS must page out the data set. If MVS must page out the data set, MVS writes it to a paging device. Programs that use VIO data sets access them just as if the data sets were allocated to a real I/O device. VIO is usually only set on for the user-defined esoteric called VIO.

Defining Switches



```
Goto Filter Backup Query Help
-----
CBPDSWF0                               Switch List           Row 1 of 6 More:  >
Command ===> _____ Scroll ===> PAGE

Select one or more switches, then press Enter. To add, use F11.

/ ID Type +          Serial-# + Description          CU    Dev
_ AA 9032           0101479032 Switch AA           001A  001A
_ AB 9032           0101469032 Switch AB           001B  001B
_ AC 9032           0106789032 Switch AC           001C  001C
_ AD 9032           0106793092 Switch AD           001D  001D
_ AE 9032-3         0209959032 Switch AE           001E  001E
_ AF 9032-5         0408699032 Switch AF           001F  001F
***** Bottom of data *****
```

Figure 87. Defining switches

3.25 Defining switches

To define switches and their associated ports, you need to:

- Define switch characteristics
- Define connections to CHPIDs, CUs, and other switches
- Define switch configuration data (port matrix)

On the primary task selection panel, select Define, modify, or view configuration data and on the resulting panel the object Switches. HCD displays the list of all switches currently defined in the IODF, as shown in the visual.

The Switch List panel (left part), lists one switch control unit and device. If there are more than one switch control unit and device, the list entry gets an indication ('>'). With the F20=Right key, you can scroll to the right part of the Switch List panel. Up to five switch control units and devices can be shown. If there are more, an indication is given for the corresponding entry ('Yes' in column 'More?' on the right part of the Switch List panel). These additional switch control units and devices can be viewed, for example, on the Port List for port FE.

Adding Switches



```
Goto  Filter  Backup  Query  Help
                                Add Switch
-----
CBDFSW10

Specify or revise the following values.

Switch ID . . . . . 01 (00-FF)
Switch type . . . . . 9032-5_____ +
Serial number . . . . . 123459032_
Description . . . . . 9032 model 5_____

Specify the port range to be installed only if a larger range
than the minimum is desired.

Installed port range . . 04 - FB +

Specify either numbers of existing control unit and device, or
numbers for new control unit and device to be added.

Switch CU number(s) . . . 0E01 _____ +
Switch device number(s) . 0E01 _____
```

Figure 88. How to define a switch

3.25.1 How to define a switch

The switch definition process is as follows:

Select **Switches** from the Define, Modify or View Configuration Data panel.

Press F11 to add a switch. This displays the panel shown in the visual.

Adding a 9032 or 9033 ESCD with the Add Switch function results in HCD optionally generating the switch's control unit and I/O device definitions used for host communication. The device can later be defined to the appropriate operating system configurations.

On this panel, HCD allows you to define the switch itself, the range of ports installed on your ESCON director, the switch device, and the switch control unit. The control unit is automatically connected to port FE and the switch device is connected to the control unit. This ensures that the definitions of switch control unit and switch device are consistent. Likewise, when deleting a switch, the switch control unit and switch device is deleted as well. However, you have still to perform the following steps:

1. Connect the switch control unit to the processor (this also establishes the switch device-processor connection).
2. Connect the switch device to the operating system.

After you have entered the new switch definition data, all defined switches are displayed in the HCD Switch List panel.

Defining Processors

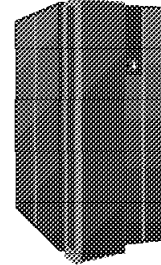


★ Basic mode

- ▶ Operating system has full use of entire machine

★ LPAR mode

- ▶ Processor Resource/System Manager (PR/SM)



```
Processor List          Row 1 of 2 More:
Command ===> _____ Scroll ===> PAGE

Select one or more processors, then press Enter. To add, use F11.

/ Proc. ID Type +  Model +  Mode+ Serial-# + Description
```

Figure 89. Defining processors

3.26 Defining processors

A CEC can run in either of two ways:

- Basic mode
- LPAR (partitioned) mode

3.26.1 Basic mode

In basic mode you run only one operating system image on the CEC and it controls all the resources (memory, channels, and so forth) of the processor.

3.26.2 LPAR mode

The Processor Resource/Systems Manager (PR/SM) feature allows a single CEC to run multiple operating system images (including the coupling facility control code (CFCC), the operating system of a coupling facility) in logically partitioned (LPAR) mode.

Each operating system has its own logical partition, which is a separate set of system resources including:

- A portion of storage (central, or central and expanded).

- One or more CPUs. The CPU can be dedicated or shared. When the CPUs are shared you assign a weight to guarantee the use for each partition of a certain CPU share or to impose a limitation (capping).
- Channels, which can be shared or dedicated.

3.26.3 Defining processors with HCD

You can define more than one processor in an IODF and for each defined processor you can configure processor-related data for further use by the channel subsystem.

HCD allows you to define and control I/O configurations for a local as well as for all other processors that are part of an S/390 microprocessor cluster.

For processors that are physically partitioned, you must define each physical partition as an individual processor.

On the primary task selection panel, select Define, modify, or view configuration data and on the resulting panel the object Processors. HCD displays the Processor List of all processors currently defined in the IODF.

Information for Defining a Processor



- ★ Processor type and model
- ★ SNA address of the support element
- ★ Serial number
- ★ Support level of the processor
- ★ Processor identifier name
- ★ Processor mode: BASIC or LPAR
- ★ Partition name
- ★ Partition number

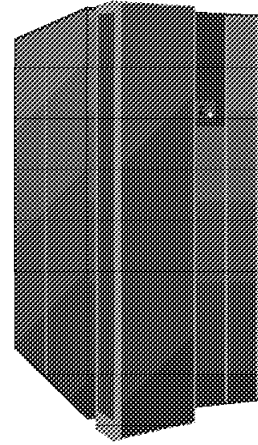


Figure 90. Information required to define a processor

3.26.4 Information required to define a processor

To define processors in your configuration, you need to know:

- *Processor ID*: An eight-byte alphanumeric name that you assign to identify the CEC in HCD, such as, BIGBLUE.
- *Support Level*: Depending on the processor type/model, there may be more than one support level for the processor type. The support level defines the supported channel path types, and the features such as dynamic I/O reconfiguration, EMIF, and coupling facility support. If the processor has several support levels, HCD displays another panel showing a list of available support levels for the processor.
- *Processor Type and Model*, such as 9672 R65
- *Configuration Mode*: BASIC or LPAR. If a processor is in LPAR mode, you must define partitions in your configuration.
- *Serial number*: If you specify a serial number, the system uses the number to verify that it is updating the correct CEC during IOCDS download.
- *Description*: Text that you use to describe the processor.
- *SNA address (Network Name) and CPC name*: For a CEC located in an S/390 microprocessor cluster, that is, several CECs connected in a token-ring hardware management console (HMC) LAN, you need the system network architecture (SNA) address of its support element in that LAN and its central processor complex (CPC, that is a synonym of CEC) name, usually the same as the processor ID.

How to Define a Processor



```

                                Add Processor
Specify or revise the following values.

Processor ID . . . . . SCZCF01_

Processor type . . . . . 9672____ +
Processor model . . . . . R76____ +
Configuration mode . . . . . LPAR +

Serial number . . . . . 0102519672
Description . . . . . _____

Specify SNA address only if part of an S/390 microprocessor cluster:

Network name . . . . . USIBMSC_ +
CPC name . . . . . SCZCF01_ +
```

Figure 91. How to define a processor

3.26.5 How to define a processor

You see in this visual the panel used to define a processor (CEC). From the Define Modify or View Configuration Data panel select **Process Processor**. This displays the list of the processors that have been defined. From the Processor list panel, press F11 to add a processor. The data entry fields are shown with sample data.

On the Add Processor panel, you can specify the network name and the CPC name, when the processor is configured in an S/390 microprocessor cluster. Use Prompt on the Add Processor panel for the SNA addresses for those CPCs that are currently configured in the S/390 microprocessor cluster.

Depending on the processor type/model, there may be more than one support level for the processor type. The support level defines the supported channel path types, and the features such as Dynamic I/O Reconfiguration, EMIF, and coupling facility support. If the processor has several support levels, HCD displays another panel showing a list of available support levels for the processor. Select the appropriate support level. HCD uses this level when validating the configuration for this processor. It relates to the installed microcode.

Work with Partitions



```
Processor List      Row 1 of 2 More:
Command ==> _____ Scroll ==>
Select one or more processors, then press Enter. To add, use F11.

/ Proc. ID Type +   Model +   Mode+ Serial-# + Description
/ SCZPF01 9672     X77      LPAR  0508229672 _____
```

```
Actions on Selected Processors
Command ==> _____
Select one or _____
_____ 1. Add like . . . . . (a)
        2. Repeat (Copy) processor configurations (r)
        3. Change . . . . . (c)
/ Proc. ID Ty 4. Prime serial number . . . . . (i)
/ SCZPF01 96 5. Delete . . . . . (d)
*****      6. Work with partitions . . . . . (p)
              7. Work with attached channel paths . . . (s)
              8. Work with attached devices . . . . . (u)
              9. View related CTC connections . . . . . (k)
              10. View processor definition . . . . . (v)
```

Figure 92. Work with partitions

3.26.6 Work with partitions

Define partitions as follows:

- On the primary task selection panel, select Define, modify, or view configuration data and on the resulting panel the object Processors. HCD displays the Processor List of processors currently defined in the IODF.
- On the Processor List panel, select the processor and the Work with partitions action from the context menu (or action code p). HCD displays the Partition List panel showing the currently defined partitions for the designated processor.

How to Add a Partition to a CEC



```

----- Partition List -----
Goto Backup Query Help
-----
Command ==> _____ Scroll ==> PAGE

Select one or more partitions, then press Enter. To add, use F11.

Processor ID . . . . : SCZCF01
Configuration mode . : LPAR

/ Partiti
*****
Add Partition
-----
Specify the following values.

Partition name . . . MVSPROD_
Partition number . . 01
Partition usage . . OS      +

Description . . . . Production MVS_____

```

Figure 93. How to add a partition to a CEC

3.26.7 How to add a partition to a CEC

If the CEC is defined in LPAR mode you now define partitions for the configuration. The 9672 supports up to 15 separate partitions. To define partitions, you need to decide on:

- Partition name, such as MVSPROD or TEST.
- Partition number, if partitions share channel paths. A partition number is used by the channel subsystem to direct I/O operations to the correct partition on shared channels.
- Whether it is to be used by an operating system, such as MVS (OS option), or by a CFCC partition (CF option).

Partitions are added as follows:

1. Select a processor from the Processor List panel and the **Work With Partitions** action from the context menu.
2. On the Partitions List panel, press F11 to add a partition, and then fill in the required information.
3. Press the Enter key. HCD will then display the updated Partition List.

Channel Types



- ★ Parallel channels (BY or BL channel paths)
- ★ ESCON channels that attach to an ESCON Converter Model 1 (CVC or CBY channel paths)
- ★ ESCON channels (CNC or CTC channel paths)
- ★ Coupling facility channels (CFS or CFR channel paths)
- ★ Integrated Cluster Bus channels (CBS or CBR channel paths)
- ★ Internal Coupling channels (ICS or ICR channel paths)
- ★ Open Systems Adapter channels (OSA, OSD, or OSE channel paths)
- ★ FICON Bridge (FCV channel paths)
- ★ FICON native (FC channel paths)
- ★ Internal DASD (2003 - 3000 models ISD)

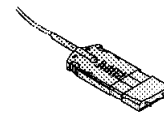


Figure 94. Channel types

3.26.8 Channel types

This visual shows the different types of channels available on an OS/390 system. A channel may be dedicated, reconfigurable, or shared depending on its type. All channel types can be defined as dedicated; in this mode the channel is assigned to only one LPAR or CEC (if running in basic mode).

3.26.8.1 ESCON channels

An ESCON channel-to-channel (CTC) channel path has an ESCON CTC connection at one end, and either an ESCON (CNC) or FICON (FCV) channel connection the other end. For any two processor complexes to communicate through an ESCON channel path, the ESCON CTC connection can be at either processor complex.

In each logical partition that can communicate with another processor complex through a shared ESCON CTC channel path, you must specify the logical address of the ESCON CTC control unit. You can specify partition numbers when defining partitions, and you can specify these partition numbers as the logical address for a CTC control unit.

3.26.8.2 FICON channel paths

FICON (fibre connection) channels increase the capacity of the channel subsystem; each FICON channel is the equivalent of eight ESCON channels.

The FCV (FICON bridge channel) channel path offers a migration path for ESCON CNC channels; using the FICON Bridge Feature on the 9032-005 ESCON director, you can attach ESCON devices to the FICON channel.

An FCV channel path occupies eight port addresses on the switch. To model the FCV bridge within HCD, consider the following: whenever you connect an eligible port address to an FCV channel path, you must set all other port addresses occupied by the FCV bridge to uninstalled.

The FC (FICON channel) channel path requires a FICON interface at the control unit. An FC channel path requires a FICON interface at the control unit, or needs to be connected to a fiber channel switch port.

Reconfigurable Channels



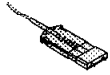
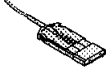
- ★ A reconfigurable channel
 - ▶ Can only be online to one partition at a time
- ★ Parallel channels (BY or BL channel paths)
- ★ ESCON channels that attach to an ESCON Converter Model 1 (CVC or CBY channel paths)
- ★ ESCON channels (CNC or CTC channel paths) 
- ★ Coupling facility channels (CFS or CFR channel paths)
- ★ Integrated Cluster Bus channels (CBS or CBR channel paths)
- ★ Internal Coupling channels (ICS or ICR channel paths)
- ★ Open Systems Adapter channels (OSA ,OSD and OSE channel paths) 
- ★ FICON Bridge (FCV channel paths)
- ★ FICON native (FC channel paths)
- ★ Internal DASD (2003 - 3000 models ISD)

Figure 95. Reconfigurable channels

3.26.9 Reconfigurable channels

This visual lists the channel types that can be defined as reconfigurable.

Reconfigurable channels are available to one logical partition at a time, but can be manually moved from one to another logical partition. Because some types of channels cannot be shared, such as parallel and CFR, you may define them as reconfigurable.

When defining a reconfigurable channel, you decide which logical partition will be assigned to it at POR time and the logical partitions that can be given access to the channel later on. This is indicated through Access and Candidate lists. Refer to 3.28, "Channel path access list" on page 146, to get more information on such lists.

Shared or EMIF Channels



- ★ ESCON channels (CNC or CTC channel paths)
- ★ Coupling facility sender channels (CFS channel paths)
- ★ Integrated Cluster Bus sender channels (CBS channel paths)
- ★ Internal Coupling sender channels (ICS channel paths)
- ★ Open Systems Adapter channels (OSA channel paths)
- ★ FICON Bridge (FCV channel paths)
- ★ FICON native (FC channel paths)
- ★ Internal DASD (2003 - 3000 models ISD)



Ficon
bridge



Escon CTC

Figure 96. Shared or EMIF channels

3.26.10 Shared or EMIF channels

The ESCON Multiple Image Facility (EMIF) provides the capability to share control units connected to ESCON channels among multiple logical partitions for a CEC operating in LPAR mode.

Though device sharing was available prior to EMIF, the only way to accomplish this was to define a separate dedicated or reconfigurable channel to each control unit from each logical partition that needed to share the associated devices. With EMIF, logical partitions can share the same device through a single shared physical channel or set of shared channels. The major advantage of EMIF is that it reduces the number of channels, which allows a cheaper and simpler I/O configuration.

With EMIF, the CEC channel subsystem provides physical path sharing by extending the logical addressing capability of the ESCON architecture to host images (PR/SM logical partitions).

Each partition has its own logical channel subsystem (logical CSS), and its own view of each shared channel (logical channel path image) and each control unit connected to the shared channel (subsystem image).

The property of being shared among logical partitions is extended to non-ESCON channels such as CF links, FICON channels, and OSA channels.

When defining the shared channel you decide which logical partition will be assigned to the channel. This is indicated through Access and Candidate lists. Refer to 3.28, "Channel path access list" on page 146, to get more information on such lists. The default with HCD for shared channels is to have all the partitions in the access list so the channel is online to all LPARs on the processor.

Information Required to Add Channels









| | | |
|------------------------------|--|---|
| ★ Type of channel |  |  |
| ★ Operation mode | Gigabit ethernet | Ficon bridge |
| ▶ Dedicated | | |
| ▶ Reconfigurable |  |  |
| ▶ Shared | CBR | CFS |
| ★ Access and candidate Lists |  |  |
| ★ Description text | Ficon native | Escon CNC |

Figure 97. Information required to add channels

3.26.11 Information required to add channels

To define channel paths in your configuration, you first make your physical layout decisions such as your CHPID numbers for channels, number of channel paths, channel path type, and switch (ESCD) information if channel path(s) are connected to a switch. The switch information includes entry switch ID, ports, and dynamic switch ID.

Then you need to decide on an operation mode (dedicated, reconfigurable, or shared) and the associated access and candidate lists.

Note: The ICONs shown are for the instances and defined channel types displayed on the 9672 hardware management console (HMC).

Work with Channel Paths



```
Processor List      Row 1 of 2 More:
Command ==> _____ Scroll ==>
Select one or more processors, then press Enter. To add, use F11.

/ Proc. ID Type +   Model +   Mode+ Serial-# + Description
/ SCZPF01 9672     X77      LPAR  0508229672 _____

Actions on Selected Processors
Select by number or action code and press Enter.

Select one or
_____
/ Proc. ID Ty
/ SCZPF01 96
*****
1. Add like . . . . . (a)
2. Repeat (Copy) processor configurations (r)
3. Change . . . . . (c)
4. Prime serial number . . . . . (i)
5. Delete . . . . . (d)
6. Work with partitions . . . . . (p)
7. Work with attached channel paths . . . (s)
8. Work with attached devices . . . . . (u)
9. View related CTC connections . . . . . (k)
10. View processor definition . . . . . (v)
```

Figure 98. Work with channel paths

3.26.12 Work with channel paths

On the primary task selection panel, select Define, modify, or view configuration data and on the resulting panel the object Processors. HCD displays the Processor List of processors currently defined in the IODF.

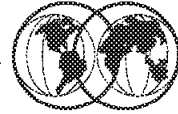
On the Processor List panel, select the processor and the Work with attached channel paths action from the context menu (or action code s). HCD displays the Channel Path List panel showing all channel paths defined for the selected processor.

Channel paths can be dedicated, reconfigurable, or shared. The following list explains when to use which channel path operation mode:

- DED** Dedicated; if you want only one logical partition to access a channel path, specify that channel path as dedicated. You cannot reconfigure a dedicated channel path. This is the default mode.
- REC** Reconfigurable; if you want only one logical partition at a time to access a channel path and you want to be able to reconfigure the channel path from one partition to another, specify that channel path as reconfigurable.
- SHR** Shared; if you want more than one logical partition to access a channel path simultaneously, specify that channel path as shared.

On the Add Channel Path panel, enter a channel path type and use F4=Prompt for the operation mode to find out the allowed operation modes for the specified type.

How to Add a Channel Path



```

                                Add Channel Path

Specify or revise the following values.

Processor ID . . . : SCZPF01          This is the main processor
Configuration mode : LPAR

Channel path ID . . . . . 80 +
Number of CHPIDs . . . . . 4
Channel path type . . . . . CNC +
Operation mode . . . . . SHR +
Description . . . . . _____

Specify the following values only if connected to a switch:

Dynamic switch ID . . . . . 01 + (00 - FF)
Entry switch ID . . . . . 01 +
Entry port . . . . . C1 +
```

Figure 99. How to add a channel path

3.27 How to add a channel path

Having defined the partitions, you can now define the channel paths.

The following example shows how to add a shared channel. This is a two step process:

- Define the channel path characteristics.
 - Define the channel path access to logical partitions.
1. Select a processor from the Processor List panel, and the **Work with attached channel paths** from the context menu.
 2. Press F11 on the channel list to add a channel path.

HCD distinguishes between the dynamic and entry switch when defining a channel path. The dynamic switch is the switch holding the dynamic connection; the entry switch is the switch to which the channel path is physically plugged.

Note: You can define multiple channels in one step. If you do so and have also specified an entry switch and entry port for the channel path, HCD displays another panel where you can specify the entry switch and port number for the subsequent channel paths.

3. You must specify the connection between channels and partitions in the subsequent panels.

Access and Candidate Lists

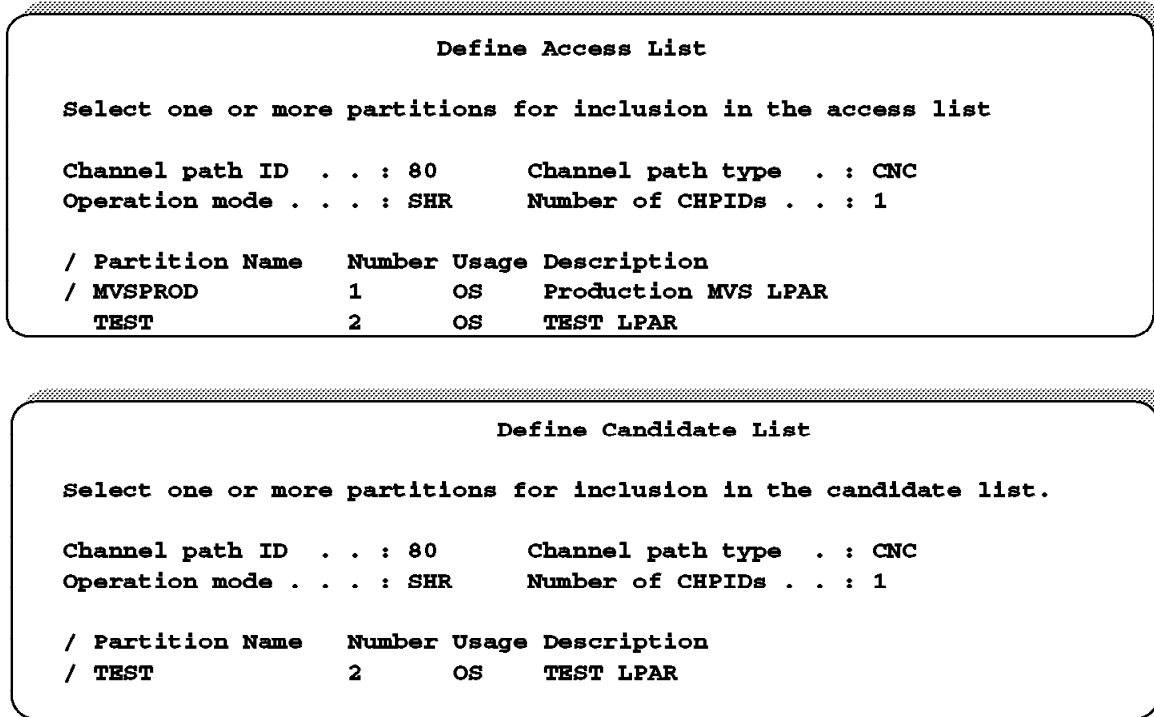


Figure 100. Access and candidate lists

3.28 Channel path access list

A logical partition that is on a channel path's access list can access the channel path when the logical partition is initially activated at POR. When a channel path is dedicated or reconfigurable, you specify one logical partition on the channel path access list. When a channel path is shared, you can specify more than one logical partition on the channel path access list.

This information is taken from the IOCDS written from the IODF that contains the definition. This information remains the same as long as no configuration changes are made while the IOCDS is active.

3.29 Channel path candidate list

A logical partition that is on a channel path's candidate list can eventually access the channel path. A logical partition on this list can access the channel path when the channel path is manually configured online to the logical partition.

HCD automatically considers a logical partition in an *access* list to be in the *candidate* list so you do not need to also enter a logical partition in the *access* list into the *candidate* list. This is valid as long as no configuration changes are made while the IOCDS is active.

In this example we will add CHPID 9C to the access list of MVSPROD and the candidate list of TEST. You use the / key to select the the partitions.

How to Add a Control Unit

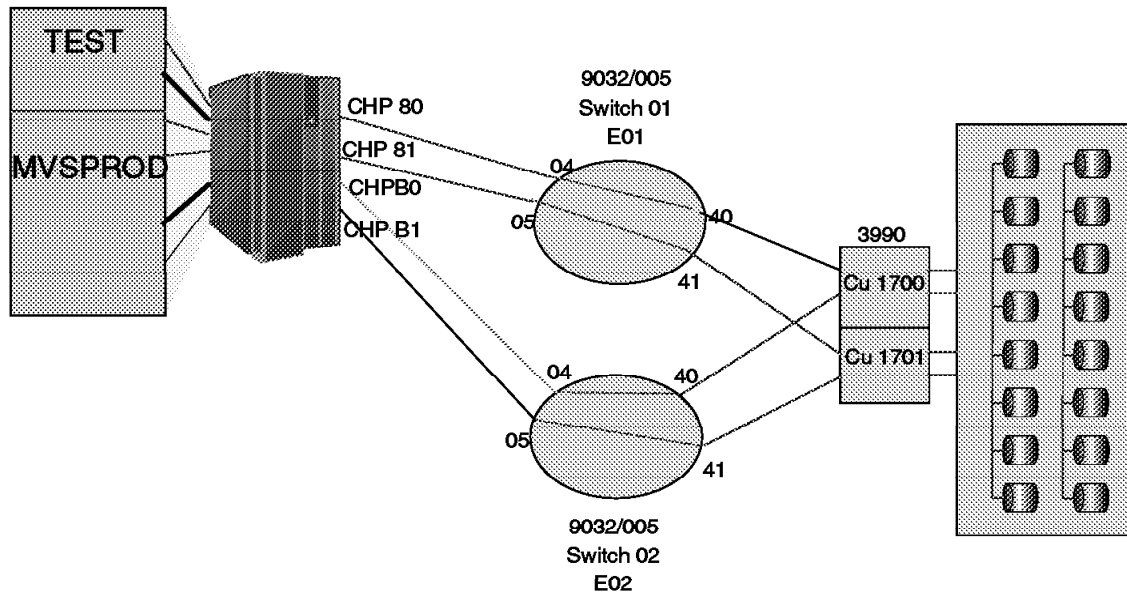


Figure 101. How to add a control unit

3.30 How to add a control unit

The configuration being defined in this example is a 3990-6 using channels attached to two ESCON directors and EMIF channels to the processor.

The steps are:

1. Define the ESCON channels to the complex, sharing them between partitions and connecting them to the ESCD switches. In this example the channels have already been defined.
2. Define the two 3990 control units, specifying the channel paths.
3. Define the DASD devices and connect them to the logical control units.

Information Required to Define a Control Unit



- ★ Control unit type and model
- ★ Control unit number
- ★ Connections to switches
- ★ Description: text
- ★ Channel path IDs
- ★ Link address
- ★ Unit address
- ★ Protocol
- ★ I/O concurrency level
- ★ CUADD value

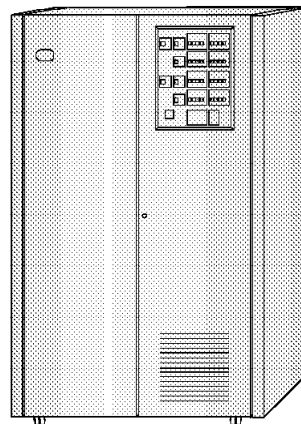


Figure 102. Information required to define a control unit

3.31 Information required to define a control unit

To define a control unit you should specify:

- Control unit type and model, such as, 9343-1
- Control unit number, such as 0060 (just for HCD correlation)
- Connections to ESCD switches
- Channel path IDs (CHPIDs) where the control unit can connect
- Link address (ESCD port numbers where the control unit is connected with the respective channel, as paired in CHIPDs)
- Control unit serial number
- Description: Which is text that you use to describe the control unit
- Processor(s) to which the control unit connects
- Information to attach the control unit to the channel path of the processor:
 - Unit address ranges that the control unit recognizes
 - Protocol (only for parallel control units)
 - I/O concurrency level: classification of a control unit based on its ability to concurrently operate and control the activity of attached devices without causing loss of control or data
 - Logical address: Which is known as the CUADD value

Adding a Control Unit



```
Goto Filter Backup Query Help
-----
CBDPCUF0                      Control Unit List                      R
Command ===> _____ Scroll ===

Select one or more control units, then press Enter.  To add, use F11

/ CU  Type +      Serial-# + Description
_ 001A 9032      0101479032 Switch AA
_ 001B 9032      0101469032 Switch AB
_ 001C 9032      0106789032 Switch AC
_ 001D 9032      0106793092 Switch AD
_ 001E 9032-3    0209959032 Switch AE
_ 001F 9032-5    0408699032 Switch AF
```

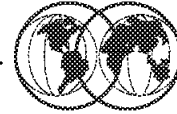
Figure 103. Adding a control unit

3.31.1 Adding a control unit

On the primary task selection panel, select Define, modify, or view configuration data and on the resulting panel the object Control units. HCD displays the Control Unit List panel showing all control units currently defined in the IODF.

Use F11=Add to define a new control unit.

Defining a 3390- 6 Control Unit



```
CBDPUCUF0          Control Unit List
Command ==> _____ Scroll ==> PAGE

Select one or more control units, then press Enter.  To add, use F11.

/ CU  Type +      Serial-# + Description
- 001 _____ Add Control Unit
- 001 CDBPCU10
- 001
- 001 Specify or revise the following values.
- 001
- 001 Control unit number . . . . 1700 +
      Control unit type . . . . . 3990-6_____ +
      Serial number . . . . . 12343990__
      Description . . . . . 1700 string_____

      Connected to switches . . . 01 02  _ _ _ _ _ +
      Ports . . . . . 40 40  _ _ _ _ _ +

      If connected to a switch, select whether to have CHPIDs/link
      addresses, and unit address range proposed.

      Auto-assign . . . . . 2  1. Yes
                           2. No
```

Figure 104. Defining a 3990-6 control unit

3.32 Defining a 3990-6 control unit

To define a 3990-6 control unit, do the following:

1. Select **Control Units** from the Define, Modify or View Configuration Data panel.
2. Press F11 to add a new control unit.
3. After typing in the details press Enter to display the Select Processor / Control Unit panel.
4. Select the processor you wish to attach the control unit to by typing a slash (/), pressing Enter, and then selecting the **S select (connect, change)** action from the context menu.
5. After filling in the details press Enter to display the Select Processor / Control Unit panel.
6. Pressing Enter validates and saves the entered data and redisplay the Select Processor / Control Unit panel, as shown in Figure 105 on page 152.
7. Press F11 to add the second control unit 1701.
8. After filling in the details press Enter to display the Add Control Unit panel.
9. Pressing Enter again returns you to the Control Unit List panel.

3.32.1.1 Defining switch connections

The Add Control Unit panel can also be used to specify the switches and ports the control unit is attached to. If you specify Yes for Auto-assign and the control unit is connected to at least one switch, HCD proposes CU-processor attachment parameters (channel path/link addresses and the unit range) based on the switch/ports the control unit is connected to. HCD will propose up to eight channel path/link address pairs, starting with the channel path that has the lowest number of devices attached

to it. On the following Select Processor / Control Unit panel you can type over the fields that are different from the proposed attachment values.

Select Processor / Control Unit



```

                Select Processor / Control Unit
CBDPCUP0                               Row 1 of 3 More: >
Command ===> _____ Scroll ===> PAGE

Select processors to change CU/processor parameters, then press Enter.

Control unit number . . . : 1700      Control unit type . . . : 3990-6

          Log. Addr. -----Channel Path ID . Link Address + -----
/ Proc. ID Att. (CUADD) + 1---- 2---- 3---- 4---- 5---- 6---- 7---- 8----
- SCZP601      ___      08.FE 28.FE _____ _____ _____ _____
- SCZP701      ___      08.FE _____ _____ _____ _____
/ SCZCF01      ___      _____ _____ _____ _____ _____
***** Bottom of data *****
```

Figure 105. Select processor for control unit

3.32.2 Select processor for control unit

After pressing the Enter key on the Add Control Unit panel, HCD displays a list panel, shown in the visual, that shows all the defined processors. You can then define how the control unit is attached to one or more processors.

A Yes in the Att. column indicates that the CU is attached to the processor.

When you place a next to the processor, you select the panel shown in Figure 106 on page 153.

Select Processor for Connections

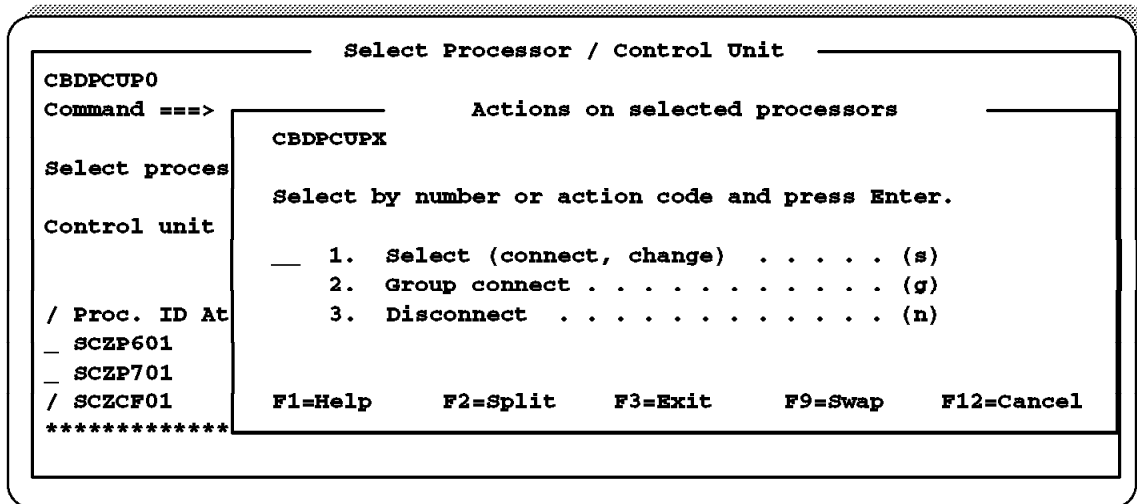


Figure 106. Defining a 3990-6 control unit

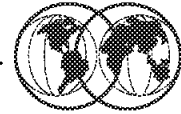
3.32.3 Defining a 3990-6 control unit

Select a processor and the Select (connect/change) action from the context menu (or action code s).

When a control unit is attached to multiple processors, you can use the Group connect action from the context menu (or action code g). This group action is particularly useful when performing symmetric changes, for example, on CPCs defined in an S/390 microprocessor cluster. The changes are applied to all selected processors when you issued the change action against a group of processors.

The next panel, shown in Figure 107 on page 154, is the result of using the options shown in the visual.

Defining Processor Attachment Data



```

Select Processor / Control Unit
-----
CBDPCUP0
C      Add Control Unit
S      CBDPCU12
S      Specify or revise the following values.
C
Control unit number . . : 1700          Type . . . . . : 3990
Processor ID . . . . . : SCZCF01
/
/ Channel path IDs . . . . 80  81  B0  B1  _ _ _ _ +
* Link address . . . . . 40  41  40  41  _ _ _ _ +
Unit address . . . . . 00  _ _ _ _ _ _ _ _ +
Number of units . . . . 32_ _ _ _ _ _ _ _
Logical address . . . . _ _ + (same as CUADD)
Protocol . . . . . _ _ + (D,S or S4)
I/O concurrency level . 2 + (1, 2 or 3)
    
```

Figure 107. Defining processor attachment data

3.32.4 Defining processor attachment data

On the Add Control Unit panel specify the channel paths that connect the control unit to the processor.

If the control unit is attached to a switch, you have to define a link address for each channel path. The link address is the port to which the control unit attaches. If the control unit attaches only to one port, the link address is the same for each channel.

You must also specify the unit address and the number of units, that is the unit address range of I/O devices that the control unit recognizes. Serial control units may have specified only one unit address range starting with 00.

If the path to the control unit is not unique, you have to specify a logical address. A logical address can be from 0 to F, that means, you can specify up to 16 serial control units for one unique path. If the control unit is not connected via a switch, the link address is ** for all control units and cannot be specified. You have to specify a logical address if more than one serial control unit connects to the same channel path ID.

Press the Enter key. HCD displays the updated Select Processor / Control Unit panel.

Repeat defining processor attachment data for all processors the control unit should be attached to.

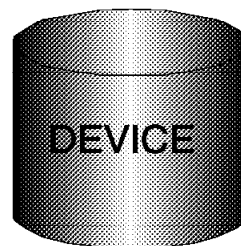
Press the Enter key to return to the Control Unit List panel.

Information Required to Define a Device



★ Information required to define a device

- ▶ Device number
- ▶ Control unit or units to which the device is to attach
- ▶ Device parameters
- ▶ Esoteric device groups



Device
number
170A

Figure 108. Information required to define a device

3.33 Information required to define a device

Before you define a device that should be defined to an operating system and to the channel subsystem (CSS), you must have defined the operating system configuration, processor, channel path, and control unit. HCD omits some steps if data is missing. For example:

- You cannot define the processor data for the device if the device is not attached to a control unit or the control unit is not attached to a processor.
- You cannot define the EDT/esoteric group data for the device until you have defined an EDT for the OS.

To define I/O devices in your configuration, you need to define:

- Device type and model, such as 3390-3.
- The device number and unit address you want assigned to the device.
- Number(s) of the control unit(s) to which the device attaches.
- Device parameters and features for defining the device to an operating system, including whether the device supports dynamic configuration or whether a tape device is automatically switchable.
- For esoteric device groups (that you named in the EDT as part of defining operating system data) which I/O devices you include in each group.
- The I/O devices that you will allow MVS to use as NIP consoles.

- Device parameters such as preferred channel path ID; for example, when SAP is selecting which channel should be tested first in order to start the I/O operation, the standard algorithm of rotate (distribute the device load by all channels reaching the device) is replaced for this device by a preferred algorithm. The same designated channel is always tried first. The preferred scheme is recommended for 3490 tapes.

S/390 Device Numbering



★ Device numbering for parallel channels

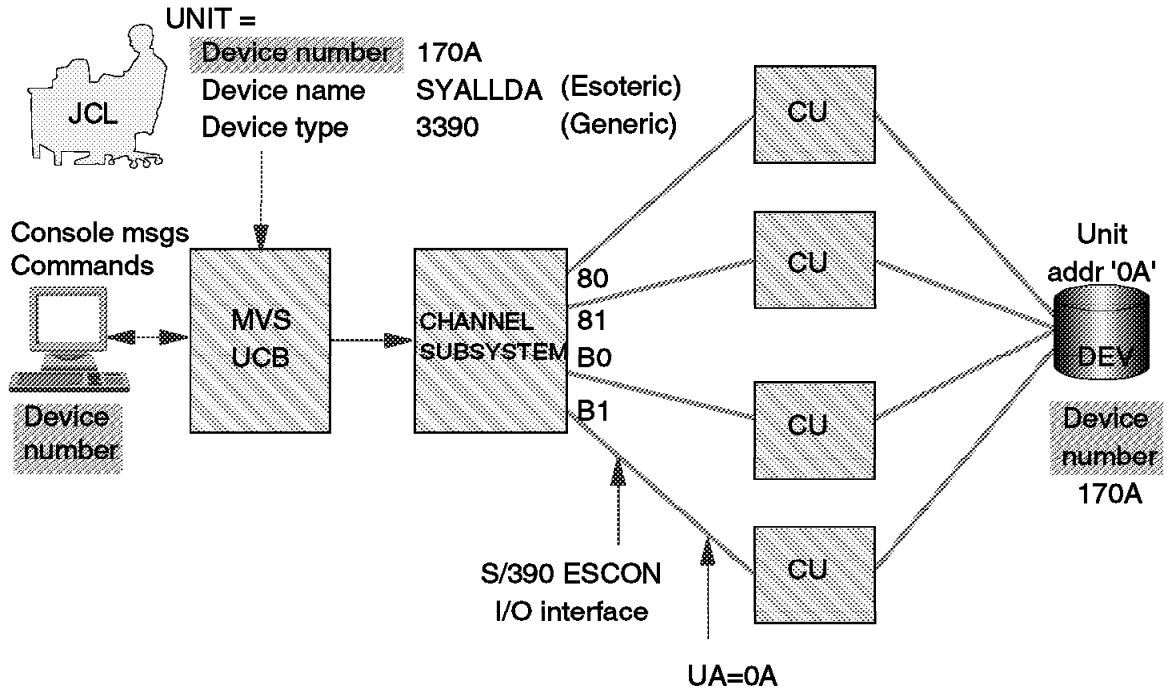


Figure 109. S/390 device numbering

3.33.1 S/390 device numbering

Operating systems need I/O device data to address the devices. The channel subsystem (CSS) also needs the data to provide the required information for performing I/O operations to a specific device.

A device number is the number you assign to identify a device in HCD. You assign a device number to each device to identify it in the configuration. A device number may be any hexadecimal number from X'0000' to X'FFFF'.

You need three steps to define an I/O device:

- Define device characteristics and control unit connection.
- Define CSS-related definitions for a device.
- Define OS-related definitions for a device (including EDT and esoteric group assignment).

Before you define a device that should be defined to an operating system and to the channel subsystem (CSS), you must have defined the operating system configuration, processor, channel path, and control unit. HCD omits some steps if data is missing. For example:

- You cannot define the processor data for the device if the device is not attached to a control unit or the control unit is not attached to a processor.
- You cannot define the EDT/esoteric group data for the device until you have defined an EDT for the OS.

3.33.1.1 Device numbers for a parallel access volume

As shown in the visual, when you define a parallel access volume in HCD, you define a control unit type that provides parallel access volume capability, for example '200A'. You define the devices for the control unit with parallel access volume device types. Base devices are defined using a base device type, for example '3390B' or '3380B'. Alias devices are defined using an alias device type, for example '3390A', or '3380A'. The device numbers are associated with unit addresses on the control unit using the 'unit address' parameter, which specifies the starting unit address for the set of devices being defined. The number of consecutive device numbers and unit addresses to be assigned is specified with the 'number of devices' parameter.

3.33.1.2 Four-digit device numbers

Since MVS/ESA SP Version 5, HCD supports the definition of four digits (numbers higher than '0FFF') for device numbers for the MVS operating system. The four-digit device numbers make it easier for large installations to use unique device numbers across their installation. The device numbers for MVS/ESA SP 4.3 and lower versions are still restricted to three digits. If you use four-digit definitions, these are ignored when you IPL or dynamically activate an MVS/ESA SP 4.2 or 4.3 system. The software products installed need to support four-digit device numbers as well.

Defining a Device



```

                                Add Device
-----
CBDFPDV10

Specify or revise the following values.

Device number . . . . . 1700 (0000 - FFFF)
Number of devices . . . . . 32__
Device type . . . . . 3390_____ +

Serial number . . . . . _____
Description . . . . . _____

Volume serial number . . . . . _____ (for DASD)

Connected to CUs . . 1700 1701 _____ +
```

Figure 110. Defining a device

3.33.2 How to define an I/O device

You can define, in one operation a group of I/O devices of the same type and with consecutive device numbers. You define the group by specifying the first device number and the number of devices in the group. Then HCD applies the definition to all devices in the group. On the I/O Device List panel you can type over the values that should be different.

HCD allows you to assign the same device number to more than one I/O device; that is, device numbers alone do not uniquely identify a device in an IODF. To clearly identify devices, HCD keeps track of each occurrence of the same device number by appending an internal suffix to the device number.

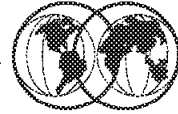
In the visual, attaching devices to the two control unit pairs completes the logical definition. The devices are defined once for all the connected control units and defined to the relevant operating system. The correlation of operating system to partition is important only when making changes to the hardware and software definitions.

The devices are defined as follows:

1. Select **Devices** from the Define, Modify or View Configuration Data panel.
2. Press F11 to add a new device. Complete the panel by specifying the device number, number of devices, device type, and control unit numbers the devices are attached to for the devices you are adding.
3. Press F11 to add the 3390 devices to the 3990 control unit pair. Complete the panel by specifying the device number, number of devices, device type, and control unit number.

4. After specifying the details, pressing Enter displays the Device/Processor Definition definition panel.
5. After confirming the responses on this panel, press Enter. You then receive the panel shown in Figure 111 on page 161.

Device / Processor Definition



```
Device / Processor Definition

Select processors to change device/processor definitions, then
press Enter

Device number . . . : 1700          Number of devices . : 32
Device type . . . : 3390

/ Processor ID   UA + Time-Out  STADET  CHPID + Preferred Explicit Device
_ SCZPF01       _      No      Yes     _      Candidate List
               _      No      Yes     _      No
```

Figure 111. Device / Processor Definition

3.33.3 Device / Processor Definition

On the Device / Processor Definition panel you can specify the CSS-related definitions by either typing over the fields in each column or by selecting a processor and pressing the Enter key. The Define Device / Processor panel is displayed.

- After confirming the responses on this panel, press Enter to display the Define Device to Operating System Configuration panel.
- Select the operating system to define the devices to, and select the **Select (connect, change)** action from the context menu (you can select all the operating systems at this time and HCD will take you through the appropriate panels in order).

Defining CSS Definitions



```

                                Define Device / Processor
-----
Specify or revise the following values

Device number . . : 1700          Number of devices . . . . : 32
Device type . . . : 3390
Processor ID . . . : SCZPF01

Unit address . . . . . 00 + (Only necessary when different from
                           the last 2 digits of device number)
Time-Out . . . . . No (Yes or No)
STADET . . . . . Yes (Yes or No)

Preferred CHPID . . . . . ___ +
Explicit device candidate list . No (Yes or No)

```

Figure 112. Defining CSS definitions for a device

3.33.4 Defining CSS definitions for a device

This visual shows the Define Device/Processor and Define Device to Operating System Configuration.

You can restrict logical partition access to an I/O device on a shared channel path by using the explicit device candidate list to select which logical partitions can access that I/O device. On the Define Device / Processor panel, enter Yes or No in the Explicit device candidate list field to specify whether you want to restrict logical partition access to an I/O device:

- YES** Specifies that only your selected logical partitions can access this I/O device. Note that the partition must also be in the channel path access or candidate list to access the device. On the Define Device Candidate List panel, place a slash (/) character to the left of each selected Partition Name.
- NO** Specifies that all logical partitions can access this I/O device. NO is the default; all logical partitions are in this I/O device's candidate list.

If you specify YES in the Explicit device candidate list field, that panel is displayed. This is done if your processor is in LPAR mode.

When you press Enter, the panel shown in Figure 113 on page 163 is displayed.

Define Device to Operating System



```
Define Device to Operating System Configuration

Select OSs to connect or disconnect devices, then press Enter.

Device number . . : 1700          Number of devices : 32
Device type . . . : 3390

/ Config. ID  Type      Description          Defined
/ MVSNEW      MVS       New configuration
_ TEST       MVS       TEST MVS  LPAR
```

Figure 113. Define device to operating system

3.33.5 Define device to operating system

After pressing the Enter key on the Define Device / Processor panel, the Device / Processor Definition panel is displayed again. Select another processor or press the Enter key again to display the Define Device to Operating System Configuration panel that shows all the defined OS configurations. You can then define the data about device parameters and features that is required by the operating system configuration.

Select an operating system and the Select (connect/change) action from the context menu (or action code s). The panel that is displayed is shown in Figure 114 on page 164.

Define Operating System Parameters



```
Define Device Parameters / Features

Command ==> _____ Scroll ==>

Specify or revise the values below.

Configuration ID . : MVSNEW           Production MVS LPAR
Device number   . . : 1700           Number of devices   : 32
Device type     . . . : 3390

Parameter/  Value   P Req.  Description
Feature
OFFLINE     No      Device considered online or offline at IPL
DYNAMIC     Yes     Device supports dynamic configuration
LOCANY      Yes     UCB can reside in 31 bit storage
ALTCTRL     No      Separate physical control unit path
SHARED      Yes     Device shared with other systems
SHAREDUP    No      Shared when system physically partitioned
```

Figure 114. Define operating system device parameters

3.33.6 Define operating system device parameters

The Parameter/Feature fields vary depending on the I/O device type and operating system type.

A plus sign (+) in the P column indicates that you can use F4=Prompt to get a list of possible values for the parameter/feature in the same row.

A Yes in the Req. field indicates that a value for the parameter/feature in the same row is required.

You accomplish the change by accepting the default values or by changing the Value entries and pressing the Enter key. The default values are set in the UIM for the device type. For parameters you can specify different default values via the OS_PARM_DEFAULT keyword in the HCD profile.

After you have defined the device parameter and feature data and pressed the Enter key, HCD displays the Assign/Unassign Device to Esoteric panel, as shown in Figure 115 on page 166.

The following device-related MVS operating system features should be reviewed:

- The OFFLINE feature defaults to the value specified in the UIM (which may not be appropriate for tape devices and ESCON directors). The offline value specified here means the device is not available in the MVS (UCB) and channel subsystem (UCW) point of view in relation to the associated MVS image.
- The DYNAMIC feature defaults to Yes for devices that allow dynamic reconfiguration. Check your subsystems and applications that manage their own devices to make sure they support dynamically modified device control blocks (UCBs).

The status of a device may be changed from DYNAMIC=YES to DYNAMIC=NO, and vice versa, by a software-only dynamic change to a configuration. When changing from DYNAMIC=NO to DYNAMIC=YES, this must be the only change made to the device at that time. You are allowed, however, to make changes to a device that is currently defined as DYNAMIC=YES and at the same time change it to DYNAMIC=NO.

Note: The DYNAMIC parameter is shown only when the appropriate device supports the dynamic I/O configuration function.

- The SHARED feature defaults to *No*. Since many MVS installations have more than one image, this parameter should be carefully checked and changed to *Yes* if appropriate

Press Enter to assign the esoterics:

1. Specify *Yes* or *No* to assign or unassign devices to the esoteric. If you do not want to assign all the devices currently being defined to this esoteric, you can limit the devices being assigned by specifying a starting device number and the number of devices to be assigned.
2. Pressing Enter returns you to the I/O Device List panel.

Assign Device to as Esoteric



```

      Assign/Unassign Device to Esoteric

Specify Yes to assign or No to unassign.  To view devices already assign
to esoteric, select and press Enter.

Configuration ID : MVSNEW           New configuration
Device number   . : 1700           Number of devices   : 32
Device type    . . : 3390           Generic . . . . . : 3390

/ EDT.Esoteric  Assigned  Starting Number  Number of Devices
_ 01.SYSDA     No        _____         _____

```

Figure 115. Assign device to an esoteric

3.33.7 Assign device to an esoteric

On the Assign/Unassign Devices to Esoterics panel, overwrite the values in the Assigned column to assign (Yes) or unassign (No) devices to the selected esoterics.

If you do not want to assign a complete group of devices, you can limit the range by specifying a starting number and the number of devices. If you omit the number of devices, 1 is assumed.

Defining a NIP Console

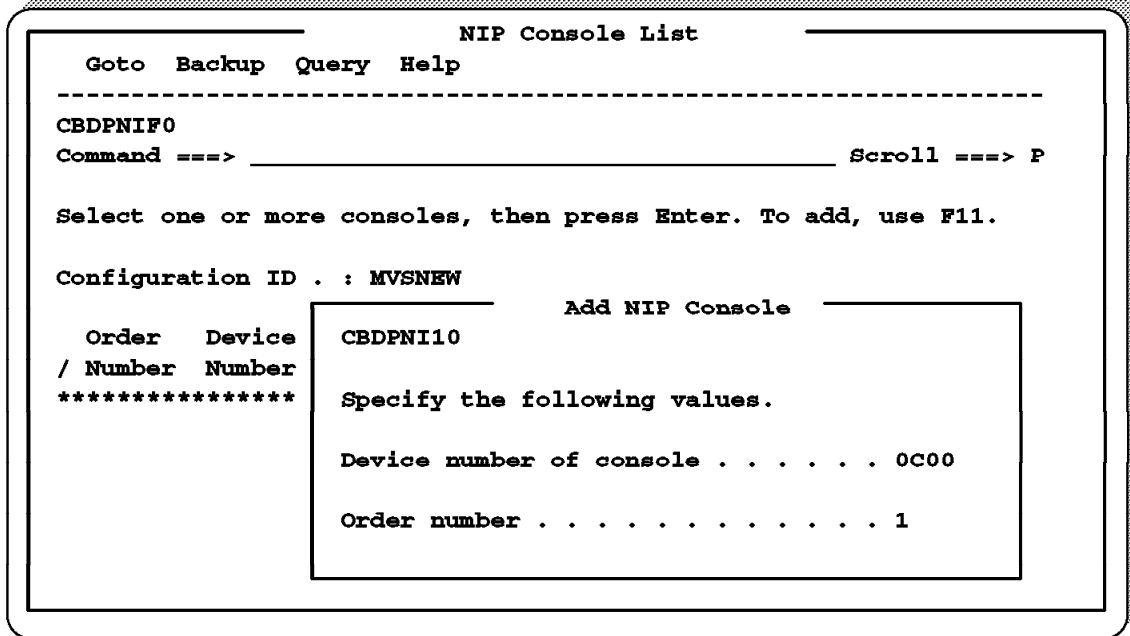
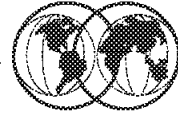


Figure 116. Defining a NIP console

3.34 Defining a NIP console

Before you can define consoles you must have defined these I/O devices to the operating system.

- On the primary task selection panel, select Define, modify, or view configuration data and on the resulting panel the object Operating system configurations. HCD displays the Operating System Configuration List panel showing all OS configurations currently defined in the IODF.
- Select an OS configuration and the Work with consoles action from the context menu (or action code n). HCD displays the NIP Console List panel (depending on the type of the selected operating system).

To define consoles, proceed as follows:

1. Define a console device and connect it to the operating system.
2. Select an Operating System Configuration and the Work with consoles action from the context menu.
3. Press F11 to add an NIP console definition.
4. On the Add NIP Console panel enter the device number of the device you want to define as a NIP console and the order number of this console. Pressing Enter again displays the panel, showing the console just defined.

Build a Production IODF



```
CBDDPM000          OS/390 Release 5 HCD
C  Activate or Process Configuration Data
  CDBDPHW20
    Build Production I/O Definition File
  S  CBDDPUT70
  S  1  Specify the following values, and choose how to continue.
  2
    Work IODF name . . . . : 'SYS6.IODF27.WORK'
    Production IODF name . 'SYS6.IODF27' _____
    Volume serial number . TOTPUB +
    Continue using as current IODF:
    2  1.  The work IODF in use at present
       2.  The new production IODF specified above
  F
  I
```

Figure 117. Build a production IODF

3.35 Build a production IODF

Although HCD validates configuration data as it is entered, a complete validation may not be performed, because data may not be defined at this time. Therefore, a “post-validation” is performed at “Build Production IODF” time. This validation might issue messages you have to deal with, according to their severity. The production IODF is not created if any errors with a severity higher than ‘warning’ are produced.

During the validation HCD invokes the IOCP program to perform checking of the channel packaging rules. Therefore, note that the correct version of the IOCP program must be accessible.

Depending on what is defined in the configuration, the work IODF must contain a definition for at least one operating system, or one processor or one switch.

- For an MVS operating system, the IODF must contain at least one EDT and one device.
- For a processor the IODF must contain a definition for at least one channel path, one control unit, and one device. If only channel path(s) of type CFR (coupling facility receiver channel path) are defined for a processor, control unit and device definition(s) can be omitted.

To build a production IODF, perform the following steps:

- On the primary task selection panel, select Activate or process configuration data.

- From the resulting panel select Build production I/O definition file, which is now shown in the visual. Enter the production IODF name and volser and HCD validates the configuration data in the work IODF.

Production IODF Created



```
CBDPM000          OS/390 Release 5 HCD
C      Activate or Process Configuration Data
      CDBPHW20

      Select one of the following tasks.

S
1  1.  Build production I/O definition file
2  2.  Build IOCDS
   3.  Build IOCP input data set
   4.  Create JES3 initialization stream data
   5.  View active configuration
   6.  Activate or verify configuration
      dynamically
   7.  Activate configuration sysplex-wide
   8.  Activate switch configuration
   9.  Save switch configuration
   10. Build OS configuration data set
F  11. Build and manage S/390 microprocessor
      IOCDSs and IPL attributes
I

      Production IODF SYS6.IODF27 created.
```

Figure 118. Production IODF created

3.35.1 Production IODF created

After the production IODF has been built, HCD informs you that the production IODF has been created.

Activating a Configuration with HCD



```
Goto Query Help
  Activate or Process Configuration Data
    Activate New Hardware and Software Configuration

Specify or revise the values for IODF activation.

Currently active IODF . . : SYS6.IODF26
  Processor ID . . . . . : SCZPF01
  Configuration ID . . . : MVSNEW           New configuration
  EDT ID . . . . .      : 01

IODF to be activated . . : SYS6.IODF27
  Processor ID . . . . . : SCZPF01  +

Test only . . . . . Yes (Yes or No)
Allow hardware deletes (FORCE, FORCE=DEVICE) . . . . No (Yes or No)
Delete partition access to CHPIDs unconditionally
(FORCE=CANDIDATE) . . . . . No (Yes or No)
Switch IOCDS for next POR . . . . . No (Yes or No)
```

Figure 119. Activating a configuration with HCD

3.36 Activating a configuration with HCD

On the primary task selection panel, select **Activate or process configuration data**, and from the resulting panel select **Activate or verify configuration dynamically**. HCD displays the **Activate or Verify Configuration** panel.

Select what you want to activate. The visual shows that you selected task **1. Activate new hardware and software configuration**. The panels when you select the other tasks are similar.

A configuration change is rejected if it includes a hardware delete for an I/O component that is online to the logical partition from which you are making the change. This is true even if you have entered YES in the *Allow Hardware Deletes* option field.

Therefore, you should vary offline any affected I/O component in all logical partitions. For example, when changing a channel path from unshared to shared, you must allow hardware deletes, and you must configure the channel path offline and vary offline the associated I/O devices before you activate the configuration.

View an Active IODF with HCD



```
OS/390 Release 5 HCD
  Activate or Process Configuration Data
    View Active Configuration
      Currently active IODF . . . : SYS6.IODF27
      Creation date . . . . . : 00-02-17
      Volume serial number . . : IODFPK
      Configuration ID . . . . . : MVSNEW      New configuration
      EDT ID . . . . . : 01
      HSA token . . . . . : SCZP601  00-02-17 17:03:38 SYS6      IODF26
      Activation scope:
      Hardware changes allowed . : Yes
      Software changes allowed . : Yes
      ENTER to view details on the activation scope.
      F1=Help      F2=Split      F3=Exit      F9=Swap      F12=Cancel
```

Figure 120. How to display active IODF from the HCD

3.37 How to display active IODF from the HCD panels

HCD allows you to view the name and status of the IODF that has been used for IPL or for the last dynamic activation. The operating system configuration and EDT identifier and, if applicable, the configuration token, which is currently active in the HSA (hardware system area), are shown. Use the view active configuration function for an overview of the actual status for dynamic activation, indicating whether hardware and software changes are allowed.

On the primary task selection panel, select Activate or process configuration data and then View active configuration.

The View Active Configuration panel is shown in the visual.

Viewing an Active IODF



- ★ View active IODF from a system console
 - ▶ D IOS,CONFIG(ALL)

```
IOS506I 18.56.28 I/O CONFIG DATA 245
ACTIVE IODF DATA SET = SYS6.IODF69
CONFIGURATION ID = L06RMVS1          EDT ID = 01
TOKEN:  PROCESSOR DATE      TIME      DESCRIPTION
SOURCE: SCZP601  99-04-21 10:00:44 SYS6      IODF69
HARDWARE SYSTEM AREA AVAILABLE FOR CONFIGURATION CHANGES
    248 PHYSICAL CONTROL UNITS
    1432 SUBCHANNELS FOR SHARED CHANNEL PATHS
    343 SUBCHANNELS FOR UNSHARED CHANNEL PATHS
    189 LOGICAL CONTROL UNITS FOR SHARED CHANNEL PATHS
    47 LOGICAL CONTROL UNITS FOR UNSHARED CHANNEL PATHS
ELIGIBLE DEVICE TABLE LATCH COUNTS
    0 OUTSTANDING BINDS ON PRIMARY EDT
```

Figure 121. How to display active IODF from an MVS console

3.38 How to display active IODF from an MVS console

You can display the active IODF and HSA usage from an MVS console with the D IOS command as shown in this visual.

How to Display a Device Status



★ Display hierarchy

- ▶ `D u,,,dddd,1` Is the device offline
- ▶ `D M=DEV(dddd)` What is the state of the paths to the device?
- ▶ `DS P,dddd` Are the paths physically available ? (DASD or tape)
- ▶ `D M=CHP(cc)` What is the state of the paths on this CHPID?
- ▶ `D M=CHP` What is the state of the CHPID?

Figure 122. How to display a device status

3.39 How to display a device status

The recommended sequence of display commands to determine the status of a device is:

1. `D U,,,ddd,1`

This command shows whether the device is online. No I/O operations are performed to present the display for this command. Therefore, the status displayed by the DISPLAY UNIT command shows the last known state of the device, and may not represent the actual physical status.

2. `D M=DEV(ddd)`

If the device is not online, it may be necessary to bring online a path to the device. This command displays the channels that are defined to access the device, and the state of the paths over those channels. The output of this command may also display PATHS NOT VALIDATED. This text means that the path status to the displayed device has not been tested.

3. `DS P,dddd`

This command displays the actual physical state of the paths to a DASD or tape device. If the path is not operational, then it is necessary to determine the state of the ESCON link and channel supporting the path.

4. `D M=CHP(cc)`

This command displays the state of the paths to devices defined as accessible by this channel.

5. `D M=CHP`

This command displays the state (online or offline) of the CHPID. Note that the display is relative to the logical partition supporting the operating system where the command is entered. Therefore, a channel shown as offline in this display may only be logically, not physically, offline.

HCD Reports



★ Two styles of reports are available with HCD

▶ Text

- CSS report
- Switch report
- OS report
- CTC connection report
- I/O path report (requires ESCON manager)
- Compare IODF

▶ Graphical

- LCU report
- CU report
- CHPID report
- Switch report
- CF connection report

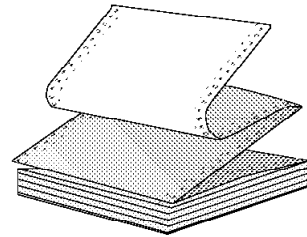


Figure 123. Producing configuration reports

3.40 Producing configuration reports

With HCD you can create and print the following reports about the configuration data in an IODF:

- Channel Subsystem (CSS) Report
- Switch Report
- Operating System (OS) Report
- CTC Connection Report
- IODF Compare Report

These reports give you a printed overview of your configurations. You can create or build reports either with HCD panels or batch jobs.

3.40.1.1 Channel Subsystem Report

The Channel Subsystem Report contains all configuration data that is used by the channel subsystem. This consists of data, in summary and detail, about your processors, partitions, IOCDs, CHPIDs, switches, control units, and I/O devices.

If you have more than one processor defined in your IODF, you can limit the report to the data for one processor or partition. When limiting the report to one partition, only those channels are reported that have the designated partition in their access list. Likewise, only control units and devices that can be reached through these channels are reported.

3.40.1.2 Switch Report

The Switch Report contains details about your switch definition, the switch configurations for each switch, and port definitions.

If your IODF contains data for more than one switch, you can limit the report to the data for one switch and the configurations for that switch.

3.40.1.3 Operating System Report

The Operating System Report contains the configuration data that is used by the OS/390 operating system. If your IODF contains data for more than one operating system, you can limit the report to the data for one operating system.

The Operating System Report function can produce three principal types of reports: The Device Report contains data about devices and has two parts:

- The Device Detail Report contains detailed data about the devices.
 - The Device Report contains summary data. The operating system summary report is not printed if you limit the OS device report to the data for one operating system.
- The EDT Report contains data about all the EDTs of your configuration.
- The Console Report contains data about all NIP consoles for MVS

3.40.1.4 CTC Connection Report

The CTC Connection Report shows CTC connections in your configuration that are defined through an ESCON director. In the case of incorrect definitions, the report contains a list of messages with diagnostic information.

If the IODF contains more than one processor or logical partition, you can limit the report to data for one processor or partition.

3.40.1.5 Compare IODFs

You can use the Compare IODFs function to compare two IODFs and report the differences between them. For greater clarity, you can limit the compare reports to certain perspectives of the IODF:

- The Processor Compare Report shows differences in the properties of partitions, CHPIDs, control units, and devices.
- The Switch Compare Report shows differences in the properties of switches and switch configurations.
- The OS Configuration Compare Report shows differences in device parameters, in features, in EDTs, in esoterics, in generics defined for EDTs, and consoles.

3.40.2 HCD graphical reports

It is possible with HCD to view and print graphical representations of your configuration. These may be stored in a data set for later printing, or viewed on a graphics-capable terminal.

Note: Prerequisite software is required for these functions. Refer to *OS/390 Hardware Configuration Definition: User's Guide*, SC28-1848, for details.

Five graphical reports may be obtained as follows:

- LCU Report

Shows the CHPIDs, control units, and devices building one or more LCUs for a designated processor.

- CU Report

Takes a control unit as focal point and shows all attachments of the control unit via switches up to the processor. It also shows the devices attached to the control unit.

- CHPID Report

Shows, for a given processor, all defined channel paths and what is attached to the CHPID (switches, CUs, and devices).

- Switch Report

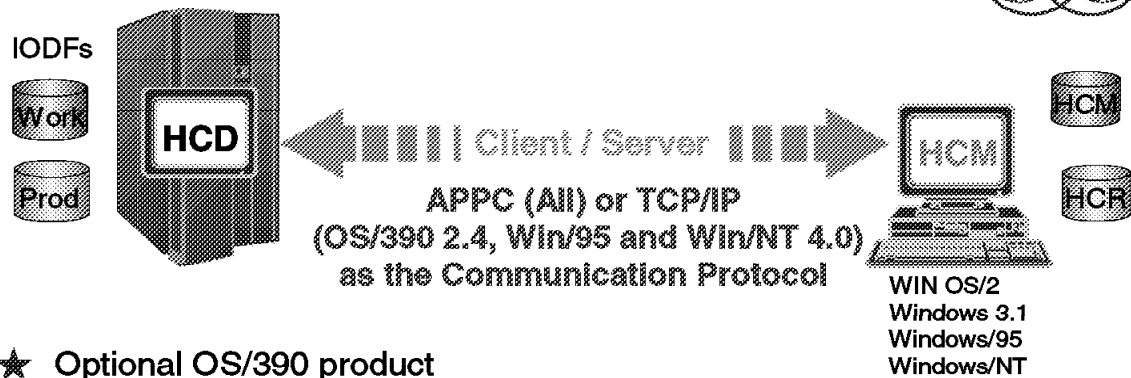
Takes a switch (ESCON director) as focal point and shows everything attached to the switch. Free ports of the switch are also shown. If the switch is connected to another switch, that switch is shown as well.

- CF Connect Report

Takes a coupling facility as focal point and shows all connections (CFS/CFR channel pairs) that exist between the coupling facility and the other processors defined in the IODF.

If *Include partitions* has been specified, the partitions are shown above each accessible CHPID.

Hardware Configuration Manager



- ★ Optional OS/390 product
- ★ Extension of HCD
- ★ PC based product
- ★ Client server to HCD
- ★ Simplifies IODF generation by using a graphical interface to define I/O configurations
- ★ Combines logical configuration data and physical data together.

Figure 124. Hardware Configuration Manager (HCM)

3.41 Hardware Configuration Manager (HCM)

HCM is an optional feature of OS/390 and extends the scope of configuration management provided by HCD.

HCM is a PC-based graphical user interface that allows you to easily navigate through the configuration diagrams and make changes in the configuration. HCM uses a client/server interface to HCD that combines the logical and physical aspects of OS/390 hardware configuration management.

All updates to your configuration are done via HCM's intuitive graphical user interface and, most important, due to the client/server relationship with HCD, all changes of the logical I/O configuration are written into the IODF and fully validated and checked for accuracy and completeness by HCD, thus avoiding unplanned system outages due to incorrect definitions.

The logical information in the IODF represents the operating system and the channel subsystem definitions; the physical information cabinets, patchports, crossbar switches, cables, locations, and so on adds the infrastructure to the logical data.

Furthermore, the logical and physical information for each object in the configuration match because they are created by the same process.

When you create an object, you add its logical and physical information at the same time. When you connect, for example, a control unit to a processor, the selected control units are logically defined to

the selected CHPID through a control unit channel interface; the physical connection, including the cable, is displayed visually in the configuration diagram.

Chapter 4. TSO/E, ISPF, and JCL

Time Sharing Option/Extensions (TSO/E) provides programming services that you can use in system or application programs. These services consist of programs, macros, and CLISTs.

TSO/E services support a wide range of functions that are useful in writing system programs as well as application programs that exploit the full-screen capabilities of TSO/E.

CLISTs, REXX execs, servers, and command processors are specific types of programs that you can write to run in the TSO/E environment.

The Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) is a set of panels that helps you manage libraries of information on the MVS system. The libraries are made up of units called *data sets* that can be stored and retrieved. You can have different kinds of information in data sets. Some examples are:

- Programs written in languages such as assembler language, COBOL, or PL/I.
- Data such as inventory records, personnel files, or a series of numbers to be processed.

For your program to execute on the computer and perform the work you designed it to do, your program must be processed by your operating system. Your operating system consists of a base control program (BCP) with a job entry subsystem (JES2 or JES3) and Data Facility Storage Management Subsystem Data Facility Product (DFSMSdfp) installed with it.

For the operating system to process a program, programmers must perform certain job control tasks. These tasks are performed through the job control statements, which are listed in this chapter. The job control tasks are introduced in the second chapter as well as introductory information about JCL. The charts in the third chapter divide these task into detailed subtasks. The tasks are:

- Entering jobs
- Processing jobs
- Requesting resources

OS/390 Facilities for System Programmers



- ★ Time Sharing Option/Extended - (TSO/E)
- ★ Interactive System Productivity Facility - (ISPF)
- ★ Job Control Language - (JCL)
- ★ Spool Display and Search Facility - (SDSF)

Figure 125. OS/390 facilities for system programmers

4.1 Installation and customization facilities

The following products are used to install and customize an OS/390 operating system:

- Time Sharing Option/Extended (TSO/E)
- Interactive System Productivity Facility (ISPF)
- Job Control Language (JCL)
- Spool Display and Search Facility (SDSF)

As a system programmer, you should be familiar with the basic tools you will use in your daily job. They are:

- TSO/E and ISPF which are used to:
 - Install and customize OS/390 and other products
 - Communicate interactively with the operating system
 - Define and maintain user definitions
 - Create data sets and JCL, and submit jobs
 - Communicate with other TSO/E users
 - Develop and maintain programs in languages such as assembler, COBOL, FORTRAN, PASCAL, PL/I, REXX, and CLIST
 - Manipulate data
- SDSF which is used to manage and monitor jobs and systems resources

- JCL which enables you to submit jobs and allocate resources

TSO/E Required customization



- ★ Define TSO/E to VTAM or TCAM
- ★ Define the users allowed to log on to TSO/E
- ★ Create TSO/E logon procedure for users

Figure 126. TSO/E required customization

4.1.1 Time Sharing Option/Extended (TSO/E)

TSO/E is a base element of the OS/390 operating system that allows users to interactively work with the system. After the required customization, users will be able to log on and issue commands from TSO/E.

Each user is defined to TSO/E by storing its user ID, logon procedure name, and the TSO/E resources which it has authority to use. This can be done in two ways:

- User Attribute Data Set (UADS), using ACCOUNT command, or
- RACF database

If RACF is installed, it can be used to control access to the system and store information about each TSO/E user. The RACF data base contains profiles for every entity (user, data set, or group) defined to RACF. For more information about the RACF, see *OS/390 Security Server (RACF) System Programmer's Guide*, SC28-1913.

As optional in the TSO/E customization, you can:

- Customize VTAM or TCAM (depending on what access methods are used in your installation):
 - VTAM: change VTAM session protocols, provide substitute characters for unavailable keyboard characters, and override default values used to start VTAM.
 - TCAM: override the default values used to start TCAM.
- Define logon limits, customizing the logon process

- Make commands and programs available
- Specify commands not supported in the background
- Specify commands and programs to run on the command/program invocation platform
- Customize HELP data
- Make host services or ISPF/PDF available
- Set performance objectives
- Monitor and protect TSO/E resources
- Customize TSO/E for different languages

For more details about customization, refer to *OS/390 TSO/E Customization*, SC28-1965.

TSO/E - TCAS start procedure



```
//TSO   PROC   MBR=TSOKEY00
//STEP1 EXEC   PGM=IKTCAS00,PARM='&MBR',TIME=1440
//PRINTOUT DD   SYSOUT=A,FREE=CLOSE
```

Figure 127. TSO/E TCAS start procedure

4.1.1.1 TCAS start procedure

Before a user can log on to TSO/E, both VTAM and the terminal control address space (TCAS) must be active in the system. The system operator enters the `START` command to start VTAM. Once VTAM has been started, the system operator enters the `START` command to start TSO/E and activate TCAS. TCAS accepts logons from TSO/VTAM users and creates an address space for each user.

The TCAS start procedure is usually stored at `SYS1.PROCLIB`. In the start procedure you specify the `TSOKEYxx` `SYS1.PARMLIB` member that contains the parameters to be used by TCAS to control the time-sharing buffers, maximum number of users, and other operational variables.

When a user logs on, the VTAM terminal I/O coordinator (VTIOC) is initialized. VTIOC controls the movement of data between TSO/E and VTAM. `SYS1.PARMLIB` member `TSOKEY00` or an installation-defined alternate member contains parameters that are used during VTIOC initialization. If a member other than `TSOKEY00` is used, the operator must include the member name either on the `START` command or in the procedure that the `START` command invokes. For a description of `TSOKEY00`, see *OS/390 Initialization and Tuning Reference*, SC28-1752.

TSO/E Logon Procedure



```
//IKJACCNT PROC
//IKJACCT EXEC PGM=IKJEFT01,
//    PARM='EX 'SYS1.OS390.CLIST(BRDCST)''',
//    DYNAMNBR=150,TIME=1440
//SYSHELP DD DSN=SYS1.HELP,DISP=SHR
//SYSPRINT DD TERM=TS
//SYSTEM DD TERM=TS,SYSOUT=*
//SYSIN DD TERM=TS
//*
```

Figure 128. TSO/E logon procedure

4.1.1.2 TSO/E logon procedure

A TSO/E logon procedure contains JCL statements that execute the required program and allocate the required data sets to enable a user to acquire the resources needed to use TSO/E. To logon to TSO/E, a user must have access to at least one logon procedure.

The logon procedure is usually located in data set SYS1.PROCLIB, or another library identified in the PROCxx concatenation in the JES2 startup procedure or in the IATPLBxx DD statement in the JES3 startup procedure. TSO/E provides a logon procedure in SYS1.PROCLIB called IKJACCNT for system programmers to access the system, for example, during the initial installation or if there are problems with the RACF data base. The foil shows a sample logon procedure. The statements specify:

PGM=IKJEFT01 Identifies the program to be executed. IKJEFT01 is the TSO/E supplied Terminal Monitor Program (TMP) that provides an interface between the user command processors, and the TSO/E control program. It obtains commands, gives control to command processors, and monitors their execution. This program can also be executed in the background by submitting JCL. Instead of the IKJEFT01 program, an installation can use the Session Manager program (ADFMDF03) or its own terminal monitor program.

PARM You can pass to IKJEFT01 a command, CLIST, REXX, or a program to be interpreted as the first line of input from the terminal after the user has logged on. In the example, it executes a CLIST or a REXX named BRDCST.

DYNAMNBR Defines the number of data sets that can be dynamically allocated at the same time. A constant of 2 is always added to the DYNAMNBR value you specify. It allows data sets to be more quickly reallocated because control blocks for data sets remain in

storage, even after the data sets have been de-allocated. You should choose the value for DYNAMNBR carefully. The value should be large enough so that it is not readily exceeded by the number of dynamic allocation requests made during the user's session. However, the larger the value you specify for DYNAMNBR the more virtual storage is used. The actual amount of virtual storage depends on the number of data sets the user allocates and de-allocates in a session. The value cannot exceed the number of concurrently-allocated resources specified in the SYS1.PARMLIB member ALLOCxx, parameter TIOT SIZE. For details, refer to *OS/390 MVS Initialization and Tuning Reference*, SC28-1752.

- SYSIN** Specifies that SYSIN is the user's terminal.
- SYSPRINT** Specifies that SYSPRINT is to be directed to the user's terminal.

Additional data sets can be allocated dynamically during the user's session or can be defined in the logon procedure. The following DD statements have special meaning and can be included in the logon procedure:

- SYSPROC** Defines the current REXX exec or CLIST library to be searched when the user uses the implicit form of the EXEC command. The visual shows the explicit form of the EXEC command and the library name is specified in the command. The implicit form would be BRDCST.
- SYSEXEC** Defines the current REXX exec library concatenation to the EXEC command when users use the implicit form of the command. By default, the system searches SYSEXEC first, followed by SYSPROC.

The data set described in SYSPROC and SYSEXEC DD statements must be partitioned, and have a record format of V, VB, F, or FB. You can allocate them dynamically using the ALLOCATE command and activate them with the ALTLIB command.

TSO/E Logon in a VTAM Environment

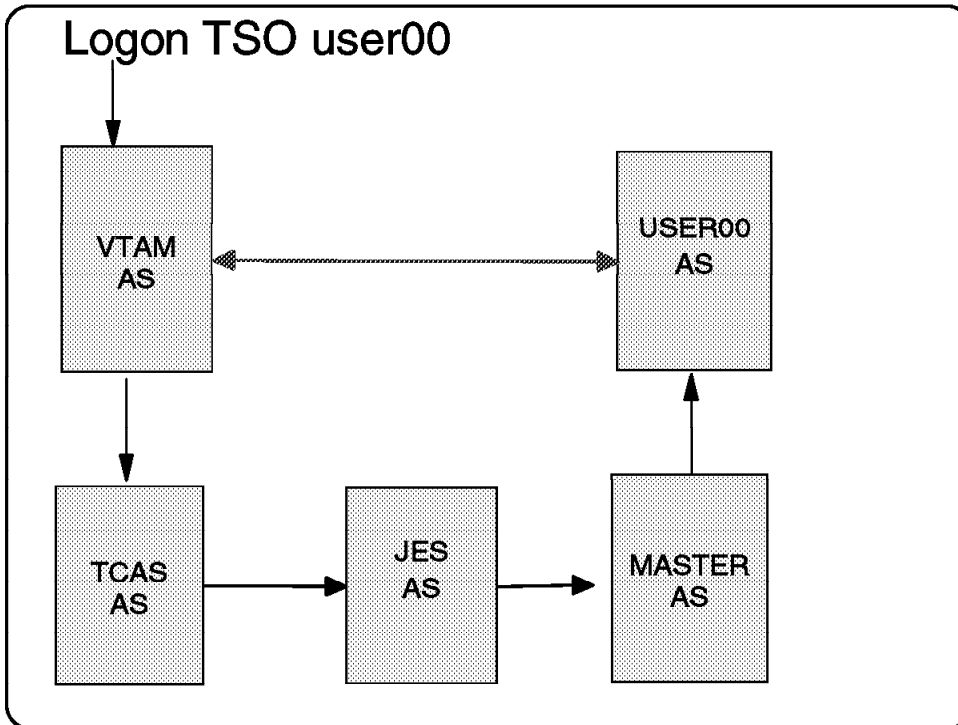


Figure 129. TSO/E logon process in a VTAM environment

4.1.1.3 TSO/E logon process in a VTAM environment

In a VTAM environment, when a user enters a LOGON command to the TSO applID:

1. VTAM receives the command and passes it to the TCAS address space.
2. If the maximum number of users logged on in the system is reached, the logon is rejected; if not, and the user ID was not specified, TCAS prompts for the user ID.
3. Once the user ID is specified, TCAS verifies that the user has authority to use TSO/E. Depending on the installation customization, a full-screen logon panel is shown to the user. Figure 130 on page 190 shows the panel displayed when the user is RACF defined. The values shown in the fields PROCEDURE, ACCT NMBR, SIZE, and COMMAND are the same the user entered for the previous TSO/E session. If the first time, they are the default values. The command entered in the COMMAND field is executed after any command entered in the PARM field on the EXEC statement of the logon procedure.
4. After the ENTER key is pressed, TSO/E verifies the values entered, then the user ID and the logon procedure name is passed to JES. The JCL is interpreted and converted. The MASTER creates the user address space and the resources specified in the JCL are allocated.
5. The user receives a screen with the READY prompt at the left top corner of the screen. This is called *line-mode TSO/E*. Now TSO/E is ready to accept commands and user interfaces, such as ISPF or SDSF can be called.

TSO/E Full-screen Logon Panel



```
----- TSO/E LOGON -----  
  
Enter LOGON parameters below:           RACF LOGON parameters  
  
Userid   ===> MYUSER  
  
Password ===>  
  
Procedure ===> IKRACONT           Group Ident ===>  
  
Acct Nmbr ===> ACCNT#  
  
Size     ===> 6072  
  
Perform  ===>  
  
Command  ===> ISPPDF  
  
Enter an 'S' before each option desired below:  
      -Nomail      -Nonotice      -Reconnect      -OIDcard
```

Figure 130. TSO/E full-screen logon panel

4.1.1.4 Using TSO/E

The visual shows that a command ISPPDF was specified. It allocates the required data sets and calls ISPF. In such cases, instead of entering TSO/E in line-mode, the user would receive the ISPF Primary Menu panel and would be in full-screen mode.

The user finishes a TSO/E session by issuing a LOGOFF command.

You probably will not use TSO/E in line-mode. The user interface provided by ISPF is a friendly way to work with TSO/E. In the following topics we will present some hints to help you when you are using TSO/E and ISPF.

For more information, refer to *OS/390 TSO/E Primer*, GC28-1967, and *OS/390 TSO/E User's Guide*, SC28-1968.

4.1.1.5 Interrupting a TSO/E function

The *Attention Interrupt* key allows you to interrupt or end a process that is taking place. If you are in a process and you want to stop or see a message requesting information you do not have, you can press the attention interrupt key to end the process.

The attention interrupt key often is labeled PA1. Sometimes it is called an escape key and is labeled Esc.

4.1.1.6 Using TSO/E as batch job

Instead of waiting at a terminal for your job to run, you can use the terminal to prepare a job containing the commands and data you would have entered at the terminal. Then use the `SUBMIT` command to run the job. In this case, you are using the facilities of TSO/E exactly as if you submitted the commands individually at the terminal. For your job, you need these job control language (JCL) statements:

- A `JOB` statement to identify your job
- An `EXEC` statement with the name of the TSO/E terminal monitor program (IKJEFT01, IKJEFT1A, or IKJEFT1B)
- At least, the following ddnames:
 1. `SYSTSPRT` which is used to control the output for your job. You can specify this DD as `SYSOUT`.
 2. `SYSTSIN` which is used as input for your TSO/E commands. It can be in stream (use an `/*` to indicate the end of the stream).
- The DDNAMEs required by the application you intend to run.

TSO/E Languages



★ CLIST

★ REXX

Figure 131. TSO/E languages

4.1.1.7 TSO/E languages

In the TSO/E environment there are two available languages: CLIST and REXX.

CLIST is an interpretative language that helps you to work more efficiently with TSO/E. It is a command list language because the most basic CLISTs are lists of TSO/E commands. When invoked it issues the TSO/E commands in sequence. The CLIST language includes the programming tools you need to write extensive, structured applications. CLISTs can perform a number of complex tasks, from displaying a series of full-screen panels to managing programs written in other languages.

The REstructured eXtended eXecutor (REXX) language is a programming language that is extremely versatile. Aspects such as common programming structure, readability, and free format make it a good language for beginners and general users. Yet because the REXX language can be intermixed with commands to different host environments, provides powerful functions, and has extensive mathematical capabilities, it is also suitable for more experienced computer professionals. The TSO/E implementation of the REXX language allows REXX execs to run in any MVS address space. You can write a REXX exec that includes TSO/E services and run it in a TSO/E address space, or you can write an application in REXX to run outside of a TSO/E address space.

CLIST and REXX can be used to customize and tailor your TSO/E environment specifically for the applications you want to use. Figure 133 on page 195 shows a sample CLIST procedure to invoke a basic ISPF environment.

ISPF Components



- ★ Dialog Manager
 - ▶ Functions
 - REXX or CLIST
 - Programs
 - ▶ Panels
 - ▶ Messages
 - ▶ Tables
 - ▶ Skeletons
 - ▶ Dialog variables
- ★ PDF
- ★ SCLM
- ★ Client/Server

Figure 132. ISPF components

4.1.2 Interactive System Productivity Facility (ISPF)

ISPF consists of four major components:

1. Dialog Manager, which provides services to dialogs and end-users. It is composed of six elements:
 - Functions: direct the dialog's processing sequence. They can be written as:
 - REXX or CLIST command procedures
 - Programs (COBOL, FORTRAN, Assembler, C/370, Pascal, PL/I, APL2)
 - Panel definitions
 - Message definitions
 - Table: two-dimensional arrays that store dialog data. They can be for temporary use, or they can be stored for use in future ISPF sessions. Not all dialogs use tables.
 - File tailoring skeletons: work like a fill-in-the-blank exercise. They take dialog variables and put them into a data set containing statements that control the output format. Some dialogs can use this kind of resource.
 - Dialog variables: pass information among dialog functions and ISPF services.
2. Program Development Facility (PDF), which provides services to assist the dialog or application developer. For example, the editor services.

3. Software Configuration Library Manager (SCLM), which provides services to application developers to manage their application development libraries.
4. Client/Server, which allows a user to run ISPF on a programmable workstation, to display the panels using the display function of the workstation operating system, and to integrate workstation tools and data with host tools and data.

Sample CLIST to Allocate ISPF and SDSF Data sets



```
ALLOC F(SYSEXEC) DS('ISP.SISPEXEC' 'ISF.SISFEXEC') REUS SHR
ALLOC F(SYSPROC) DS('USER.CLISTS.DSN' 'ISP.SISPCLIB') REUS SHR
ALLOC F(ISPPLIB) DS('ISP.SISPSENU' 'ISF.SISFPLIB') SHR
ALLOC F(ISPMLIB) DS('ISP.SISPMENU' 'ISF.SISFMLIB') SHR
ALLOC F(ISPSLIB) DS('ISP.SISPSENU') SHR REUS
ALLOC F(ISPTLIB) DS('ISP.SISPTENU' 'ISF.SISFTLIB') SHR REUS
ALLOC F(SYSHLP) DS('ISP.SISPHLP' 'ISF.SISFTLIB') SHR REUS
ALLOC F(ISPTABL) DS('USERID.PROFILE.DSN') SHR
ALLOC F(ISPPROF) DS('USERID.PROFILE.DSN') SHR
ALLOC F(ISPLIST) SYSOUT(A) LRECL(121) RECFM(F B A)
ALLOC F(ISPLOG) SYSOUT(A) LRECL(121) RECFM(V A)
ALLOC F(ISPCTL0) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(80) +
    BLKSIZE(800) RECFM(F B)
ALLOC F(ISPCTL1) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(80) +
    BLKSIZE(800) RECFM(F B)
ALLOC F(ISPCTL2) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(80) +
    BLKSIZE(800) RECFM(F B)
ALLOC F(ISPWRK1) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(256) +
    BLKSIZE(2560) RECFM(F B)
ALLOC F(ISPWRK2) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(256) +
    BLKSIZE(2560) RECFM(F B)
ALLOC F(ISPLST1) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(121) +
    BLKSIZE(1260) RECFM(F B A)
ALLOC F(ISPWRK2) NEW UNIT(VIO) SPACE(1,1) CYLINDERS LRECL(121) +
    BLKSIZE(1260) RECFM(F B A)
ISPSTART PANEL(ISR@PRIM) NEWAPPL(ISR)
```

Figure 133. Sample CLIST to start ISPF environment

4.1.2.1 ISPF required data sets

ISPF runs in a TSO/E environment. Before ISPF can be started, some data sets must be available; this can be done by allocating them in the logon procedure or dynamically using the TSO/E ALLOC command. The required DD names are:

- SYSEXEC for REXX, or SYSPROC for CLIST and/or REXX data sets
- ISPPLIB for panel definition libraries
- ISPTLIB for input table libraries
- ISPMLIB for message libraries
- ISPPROF for the user profile. ISPF and many other products installed under TSO/ISPF use this data set for storing variables and settings to be used from one TSO/ISPF session to another.

The DD names ISPTABL, for output tables, and ISPSLIB for skeletons may be requested for some specific applications.

For some data sets, if not pre-allocated, ISPF allocates them:

- ISPLOG DD name, is the data set where ISPF logs commands issued by the user and some ISPF functions log errors detected.
- ISPLIST DD name, ISPF uses this data set when user requests printed output.

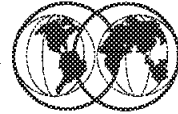
The DD names ISPCTLx, where x can be 1–9, A – W, are used by ISPF as a temporary data set. ISPF can use one for each logical screen to generate JCL or utility control statements or to generate

listings. ISPF can run up to 32 logical screens at one time. The default value is 8. The installation can change the default value by modifying the ISRCONFIG table. ISPCTL0 is used only by Edit for the SUBMIT command.

The DD names ISPWRKx are used by ISPF for file tailoring services with ISPFIL0 allocated to a PDS. The DD names ISPLSTx are used for generated listings. Refer to Figure 133 on page 195 for a JCL sample that can be inserted into a logon procedure. The same pre-allocation can be done by the ALLOCATE command in a CLIST or in a REXX exec to be executed before the ISPF start.

Figure 133 on page 195 shows a sample CLIST for allocating temporary data sets for the use of two logical screens.

ISPF Primary Option Menu



```
Menu Utilities Compilers Options Status Help
-----
                    ISPF Primary Option Menu
Option ---->

0 Settings      Terminal and user parameters      User ID . : MYUSER
1 View          Display source data or listings              Time . . . : 14:03
2 Edit          Create or change source data          Terminal. : 3278
3 Utilities     Perform utility functions                    Screen. . : 1
4 Foreground    Interactive language processing             Language. : ENGLISH
5 Batch         Submit job for language processing              Appl ID . : ISR
6 Command       Enter TSO or Workstation commands              TSO logon : IKJACCT
7 Dialog Test   Perform dialog testing                        TSO prefix: MYUSER
8 LM Facility   Library administrator functions              System ID : MVS1
9 IBM Products IBM program development products    MVS acct. : ACNT#
10 SCLM        SW Configuration Library Manager             Release . : ISPF 4.
11 Workplace    ISPF Object/Action Workplace
```

Figure 134. ISPF Primary Option menu

4.1.2.2 ISPF Primary Option Menu

ISPF is started in a TSO/E environment through an ISPF, or PDF, or ISPSTART command. The ISPF Primary Option Menu contains the options that you can use to create your own applications online. If your installation has a customized ISPF Primary Option Menu, the menu might not contain all of options shown in this visual, or it might contain certain installation-specific options.

4.1.2.3 ISPF Help function

ISPF has a powerful Help function which you can use to learn how to use all default ISPF options. You access it by pressing the Help program function key, or typing help in the COMMAND line. You can see and change the program function key assignments by using the KEYS command.

You can learn how to use ISPF by just using HELP. At the Primary Option Menu press Help to learn about the options. In the Help panel, press Help again to learn how to move through help panels. Some Help options are cursor sensitive, you have to move the cursor to that option to get more information.

The *Interactive System Productivity Facility Getting Started*, SC34-4440, provides a good source of information for beginners in the ISPF environment.

Allocating Data Sets With ISPF



★ Use ISPF Option 3.2

▶ Choose data set name

- Data set names can be up to 44 characters
- Names have qualifiers separated by periods
- SYS1.PARMLIB
- MYUSER.PRIVATE.JCLLIB
- Determine record format

▶ Allocate space for data set

Figure 135. Allocating data sets with ISPF

4.1.2.4 Allocating a data set

A data set, or file, is an area that is reserved on either disk, or tape, to enable you write programs and store data. Before you can edit or store data, you must instruct the system to allocate some space on disk, often referred to as Direct Access Storage Devices (DASD), and provide information to identify the format of this data set.

ISPF Option 3.2 enables you to reserve some space on a storage device and identify this space with a data set name, often referred to as a *DSN* or *DSNAME*.

Data set name standards are site dependant, and may be protected by RACF or another security product. If you do not have the authority to allocate a DSN with a specific name, your request will fail and you will receive an error message. A data set name can be 44 characters in length, and each qualifier must be separated by a period (.) and must not be greater than eight characters. For example:

- SYS1.PARMLIB
- SYS1.PARMLIB.BACKUP.D99156
- MYUSER.PRIVATE.JCLLIB
- MYUSER.TEST.NEW.SYSTEM.JCLLIB

The High Level Qualifier (HLQ) is the first part of the DSN. In the examples, the HLQs are SYS1 (usually reserved for MVS system DSNs), and MYUSER, which could be your user ID. Some system data sets must be named as specified, but personal or in-house data sets should be named to be

meaningful and easy to associate with a user or application, and to enable efficient security and DASD maintenance strategies to be maintained.

ISPF 3.2 Option Panel



Option ==>

| | |
|-----------------------------------|---------------------------------------|
| A Allocate new data set | C Catalog data set |
| R Rename entire data set | U Uncatalog data set |
| D Delete entire data set | S Data set information (short) |
| blank Data set information | M Allocate new data set |
| | V VSAM Utilities |

ISPF Library:

Project .. MYUSER
Group ... PRIVATE
Type JCLLIB

Other Partitioned, Sequential or VSAM Data Set:

Data Set Name ...
Volume Serial ... (If not cataloged, required for option "C")

Data Set Password .. (If password protected)

Figure 136. ISPF Option 3.2 panel

4.1.2.5 Allocating a data set

The visual shows the Data Set Utility panel that appears when you choose ISPF option 3.2.

Let us suppose you want to create a data set with the same attributes as SYS1.PARMLIB. Just specify the name of the model data set:

Other Partitioned, Sequential or VSAM Data Set:
Data Set Name . . . 'SYS1.PARMLIB'

Imbed DSN in single quotes, otherwise ISPF will add your TSO prefix as HLQ. To know your TSO prefix, enter TSO PROFILE in the COMMAND field. If NOPREFIX appears, you do not need imbed the DSN in quotes.

After entering the model data set name with blanks in the COMMAND line, ISPF displays a panel with information about the model data set and stores this information so you can use it to allocate a new data set. Pressing the ENTER key causes the previous panel to be shown. On the COMMAND line, enter an A and specify:

- A data set name in the ISPF Library:

Project . . Group . . . Type

Any data set name with a three-level name can be entered in the Project, Group, and Type fields, with one level of the name in each field. ISPF does *not* add your prefix as HLQ.

- A data set name and optionally a volume:

Other Partitioned, Sequential or VSAM Data Set:

Data Set Name . . . 'HLQ.XXX.XXXX.XXX' Imbed in single quotes,
unless you want the HLQ
to equal your prefix

If both a library and a data set name are specified on the same panel, the data set name takes priority. Therefore, to specify a library, leave the Data Set Name field blank.

Allocate New Data Set Panel



```
Allocate New Data Set

Command ==>

Data Set Name . . . : MYUSER.PRIVATE.JCLLIB

Management class . . . (Blank for default management class)
Storage class . . . . (Blank for default storage class)
Volume serial . . . . TOFTS3 (Blank for system default volume) **
Device type . . . . . (Generic unit or device address) **
Data class . . . . . (Blank for default data class)
Space units . . . . . TRKS (BLKS, TRKS, CYLS, KB, MB, BYTES
or RECORDS)

Average record unit (M, K, or U)
Primary quantity . . 30 (In above units)
Secondary quantity 15 (In above units)
Directory blocks . . 100 (Zero for sequential data set) *
Record format . . . . FB
Record length . . . . 80
Block size . . . . . 3120
Data set name type : PDS (LIBRARY, HFS, PDS, or blank) *
(Y Y/MM/DD, YYYY/MM/DD
Expiration date . . . YY.DDD, YYYY.DDD in Julian form
Enter "/" to select option DDDD for retention period in days
Allocate Multiple Volumes or blank)

( * Specifying LIBRARY may override zero directory block)
```

Figure 137. Allocate New Data Set panel

4.1.2.6 Allocate New Data Set panel

After pressing the ENTER key, the Allocate New Data Set panel is displayed. This panel shows the information ISPF has stored. You can specify new or change the data set requirements.

In this example we will allocate a partitioned data set (PDS). This type of data set allows individual members within the data set. Most environments now utilize DFSMS/MVS to control data set allocation; therefore, it is not necessary to specify management class, storage class, or data class information. In most cases the defaults will be satisfactory. We must, however, specify the following:

- Space units** Identifies whether the allocation will be in blocks (BLKS), tracks (TRKS), cylinders (CYLS), KB, MB, bytes, or records.
- Primary quantity** Identifies the amount of primary space you want to allocate in relation to the space units previously identified.
- Secondary quantity** Identifies the space that can be allocated for secondary extents, if the primary quantity fills up. Non-VSAM data sets can have a maximum of 16 extents, which includes up to five multiple extents that may have been used to satisfy the primary extents.
Note: The exception to the 16 extent limitation are partitioned data set extended (PDSE) data sets which can have up to 123 extents.
- Directory blocks** Must be specified for partitioned data sets (PDS). A directory is an index used by the system to locate members in the partitioned data set. It consists of 256-byte records, each containing directory entries. There is one directory entry for each

member. The directory is written at the beginning of the primary space. It must fit in the first extent of the data set. For partitioned data sets (PDS) ensure you request enough directory space allow for growth of the data set. You cannot lengthen the directory once the data set is created. If the directory runs out of space, you must recreate the data set.

Note: The number of member entries that fit in a directory block is as follows:

- For a data set with ISPF statistics: six entries per block
- For a data set without ISPF statistics: 21 entries per block
- For a load module data set: four to seven entries depending upon attributes.

Record format Can be any valid combination of the following:

- F - Fixed length records
- V - Variable length records
- U - Undefined format records
- B - Blocked records
- A - ASA printer control characters
- M - Machine code printer control characters
- S - Standard (for F) or spanned (for V) - sequential data sets only
- T - Track-overflow feature.

The option we will use for a partitioned data set is **FB**

Record length The logical record length, in bytes, of the records to be stored in the data set. In the case of a JCL or program library this value is 80 bytes.

Block size The block size (physical record length), in bytes, of the blocks to be stored in the data set. If records are specified, the block size specifies the average record length.

After pressing the PF3 exit key, the successful response to the allocation is indicated on the Data Set Utility panel that reappears. You are returned to this after processing the Allocate New Data Set panel. The top right of the screen will indicate *Data set allocated*.

You have now created a partitioned data set with the name, *MYUSER.PRIVATE.JCLLIB*. This is an empty data set, so the first thing you will do is add a member to this data set. See 4.1.2.7, "Adding a member to a new PDS" on page 204.

Edit Entry Panel - Option 2



```

                                     Edit Entry Panel
Command ===>

ISPF Library:
Project . . .
Group . . . . . . . . . . . . . . . .
Type . . . .
Member . . . . . (Blank or pattern for member selection)

Other Partitioned or Sequential Data Set:
Data Set Name . . . 'MYUSER.PRIVATE.JCLLIB(IEFBR14) '
Volume Serial . . . (If not cataloged)

Workstation File:
File Name . . . . .

Options
Initial Macro . . . . / Confirm Cancel/Move/Replace
Profile Name . . . . . Mixed Mode
Format Name . . . . . Edit on Workstation
Data Set Password . . Preserve VB record length
```

Figure 138. Edit Entry panel - option 2

4.1.2.7 Adding a member to a new PDS

You can add a member to a new PDS using ISPF Option 2 - Edit. When you choose this option, the Entry Edit panel is displayed. As discussed previously, if your data set has three qualifiers, you can use the *Project/Group/Type* fields to identify your data set. The advantage is that ISPF stores this information in your profile and the next time you enter this panel the fields are set by the saved information. If your data set does not have three qualifiers, you must use the *Other Partitioned or Sequential Data Set* field, embedding the data set name in single quotes, unless you want ISPF to add your prefix as an HLQ. It is necessary to specify a member name. In the example in the visual, IEFBR14 is the member being created.

ISPF Edit Panel



```
EDIT          MYUSER.PRIVATE.JCLLIB(IEFBR14) - 01.00          Columns 00001
Command ===>          Scroll ===> CSR
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
```

Figure 139. ISPF Edit panel

4.1.2.8 Editing a data set

You use the *Edit* function to create, display, or change data stored in partitioned or sequential data sets. When the editor displays existing data, each line consists of a six-column *Line Command* field followed by a 72-column data field. To view data that is not displayed, use the scroll commands. The following are PDF default values:

- F7/19 Scrolls up F10/22 Scrolls left
- F8/20 Scrolls down F11/23 Scrolls right

In Edit and View mode you can issue line commands and primary commands.

ISPF - Edit: Some Line commands



| Command | Description |
|---------|---------------------|
| I | Insert lines |
| D | Delete lines |
| R | Repeat lines |
| C | copy lines |
| M | Move lines |
| A | After line |
| B | Before line |
| (| Shift right columns |
| < | Shift right data |
|) | Shift left columns |
| > | Shift left data |
| X | Exclude lines |

Figure 140. ISPF edit - some line commands

4.1.2.9 Line commands

Line commands affect only a single line or block of lines. You enter line commands by typing them in the line command field and pressing Enter. With line commands you can:

- Insert or delete lines
- Repeat lines
- Rearrange lines or overlay portions of lines
- Simplify text entry and formatting
- Define an input mask
- Shift data
- Include or exclude lines from the display
- Control tabs and boundaries for editing
- Convert some types of special temporary lines to data lines.

The visual shows some line commands you can use. To learn about line commands, type ? in the line command field and press Enter; this causes a short Help message to appear at top right of the screen. Pressing HELP again causes a long message to appear. Pressing HELP again causes a help panel with all the line commands to appear.

ISPF Edit Panel - Inserting Lines



```
Command ==>                               Scroll ==> CSR
***** Top of Data *****
150100
***** Bottom of Data *****
```

Results in:

```
***** Top of Data *****
000100
.....
.....
.....
.....
.....
***** Bottom of Data *****
```

Figure 141. ISPF Edit panel - inserting lines

4.1.2.10 Inserting lines

The visual shows an example of the line command *Insert*. You type I to insert one line or Ixx, where xx is the number of lines you want to insert.

ISPF Edit - Repeating and Deleting Lines



```
Command ==>                               Scroll ==> CSR
***** Top of Data *****
000100 This is a test program
D00200 Issue the D command here.
***** Bottom of Data *****

Results in
***** Top of Data *****
R50100 This is a test program
***** Bottom of Data *****

Issuing the R5 command to repeat the line 5 times. Results:
***** Top of Data *****
000100 This is a test program
D00200 This is a test program
000300 This is a test program
000400 This is a test program
000500 This is a test program
D00600 This is a test program
***** Bottom of Data *****

Issuing the DD at line 000200 and again at line 000600
Results in
***** Top of Data *****
000100 This is a test program
***** Bottom of Data *****
```

Figure 142. ISPF edit - repeating and deleting lines

4.1.2.11 Repeating and deleting lines

You can delete lines by issuing the line command D which will *Delete* one line, or Dxx, where xx is the number of lines to delete. You can also delete a block of lines by using DD at the beginning and at the end of the block.

Copying lines can be done by issuing C for copying one line, or Cxx, for copying xx lines, or indicate with CC the beginning and the ending of lines to be copied. After indicating the lines, you must tell the Editor where to copy the lines to. This is done by typing either a B (for before) or an A (for after) on the line following or preceding where you want the data copied.

ISPF Edit - Save and Cancel



★ Saving your file

- ▶ Enter **SAVE** on the command line
- ▶ Use PF3 to save the file with profile option **AUTOSAVE ON**

★ Canceling updates to a file

- ▶ Enter **CANCEL** on the command line

Figure 143. ISPF edit - save and cancel

4.1.2.12 Primary commands

Primary commands affect the entire data set. You type them on the Command ==> field. You use a primary command to:

- Control your editing environment
- Find a specific line
- Find and change a character string
- Combine several members into one
- Split a member into two or more members
- Submit data to the job stream
- Save the edited data or cancel without saving
- Sort data
- Delete lines
- Access dialog element models
- Run an edit macro.

Type the **PROFILE** primary command to see the profile options you are using to edit the data set. The **RECOVERY ON** option permits you recover the changes you made before a session failure. This option creates a copy of all changes you did after a **SAVE** command. When a session fails, the next time you

return to the TSO/ISPF Edit function, you can edit the same data set with all updates and you can continue editing and save or cancel.

The UNDO command allows you to remove changes to a member during an edit session. Each UNDO issued removes the change performed since the previous ENTER was done. Subsequent UNDO commands remove previous updates in sequence from the newest change to the oldest that have occurred during this edit session. UNDO can only be used with the RECOVERY ON, SETUNDO REC, or SETUNDO STG profile options. You can prepare profiles, give them names, and choose the appropriate profile to edit a data set by typing its name at the EDIT Entry Panel PROFILE field.

Saving new or updated files: After updating:

- To save your data issue the SAVE primary command. With the profile option *AUTOSAVE ON*, pressing the F3/F15 END key ends the edition and saves the data.
- Canceling your updates can be done by issuing the CANCEL primary command. This removes all changes made since the last SAVE was performed.

JCL Streams and Jobs



- ★ Programmers create jobs
 - ▶ Using JCL statements

- ★ Jobs are submitted to and processed by:
 - ▶ JES2 or JES3
 - ▶ OS/390 operating system

- ★ Jobs have one or more job steps

Figure 144. JCL streams and jobs

4.1.3 JCL streams and jobs

For the operating system to process a program, system programmers or application programmers must perform certain job control tasks. These tasks are performed through the job control statements, which consist of:

- JCL statements
- JES2 control statements
- JES3 control statements

The operating system performs many job control tasks automatically. You can influence the way your job is processed by the JCL and Job Entry Subsystem (JES) parameters you code. The job entry subsystem you will be using will be either JES2 or JES3.

Before you begin to create a JCL stream, which will be submitted and processed by the operating system as what is commonly called a *job*, or a *batch job* you might be wondering; *What exactly is a job?*

You enter a program into the operating system as a job step. A job step consists of the job control statements that control execution of a program and request the resources needed to run the program. A job step is identified by an *EXEC* statement.

A job is a collection of related job steps and can contain up to 255 steps. A job must have:

- A *JOB* statement that assigns a name to the job and marks its beginning. The *JOB* statement is also used to provide some administrative information like security, accounting, and identification. Every job has one and only one *JOB* statement.
- An *EXEC* statement marks the beginning of a job step, assigns a name to the step, and identifies the program or procedure to be executed in the step. You can add various parameters to the *EXEC* statement to customize the way the program executes. Every job has at least one *EXEC* statement.

A job can contain *DD* (Data Definition) statements to identify and describe the input and output data sets to be used in the step.

Creating jobs - An Introduction to JCL



- ★ Create a member IEFBR14 using ISPF edit
- ★ Creating JCL statements
 - ▶ Job card statement
 - ▶ EXEC statement
 - ▶ Job steps

```
000100 //MYUSER01 JOB (ITSO),'IEFBR14',CLASS=A,MSGCLASS=X
000200 //*-----This is a comment line -----
000300 //STEP1 EXEC PGM=IEFBR14
```

Figure 145. Creating jobs - an introduction to JCL

4.1.3.1 Creating data sets using JCL statements

You can use JCL statements to create, delete, catalog, or uncatalog data sets. This visual shows the JCL we will use to create a data set and to review some of the fundamental JCL statements. For more details refer to *OS/390 MVS JCL Reference*, GC28-1757.

The JCL statements:

- Start in position one, and are identified by a // at the beginning of the line
- The commentary lines are identified by a // at start of the line
- Are coded from column 1 to column 71
- A comma indicates that the statement has continuation
- A continuation of a statement must *start* from column 4 to 16
- // and rest of statement with blanks indicates the end of the job

Comments must be separated from operators/parameter by at least one blank.

```

EDIT      MYUSER.PRIVATE.JCLLIB(IEFBR14) - 01.00      Columns 00001
Command ==>                                         Scroll ==> CSR
***** ***** Top of Data *****
000100 //MYUSER01 JOB (ITSO),' IEFBR14', CLASS=A,MSGCLASS=X
000110 /*-----This is a comment line -----

```

Figure 146. The job statement explained

JOB Statement (Job Card): The first line of the IEFBR14 member is the job statement. It specifies parameters to be used by the job entry subsystem to schedule this job for processing. The format of the job card, and the importance of the data specified in the job card vary from installation to installation. The following fields are important:

- jobname** The first field is the jobname, in this case MYUSER01. It can have up to eight characters. Some sites perform security checking against the job name to ensure standards, usually the ID of the user who submitted the job. Let us suppose the standard is: user ID suffixed with at least one alphanumeric character. If MYUSER is the user ID, MYUSER01 matches the standards. Other sites might not have any job name restrictions.
- JOB** Identifies the job to the system, when submitted. It must be present, must follow the jobname and there must be at least one space between them.
- Account** Some sites use this field for accounting and job processing information. In the example, the value is (ITSO).
- programmer name** The next field is the programmer name field, which you can use to identify the member name, for example. In the example, 'IEFBR14' is the member name.
- job class** The CLASS= field identifies the JES job class this job will execute under. In the example, CLASS=A is the JES job class. Many sites do not use this option, and the JES class is set according to your user ID. Job classes are set up at JES initialization.
- msgclass class** MSGCLASS= assigns the job log to an output class. The output class and its characteristics are identified in a parameter file used at JES initialization.

EXEC Statement: Figure 147 shows an example of the EXEC statement.

```

000100 //MYUSER01 JOB (ITSO),' IEFBR14', CLASS=A,MSGCLASS=X
000200 /*-----This is a comment line -----
000300 //STEP1 EXEC PGM=IEFBR14

```

Figure 147. The EXEC statement

In the EXEC you specify:

- Step name** In the example STEP1 gives a name to the step. We could have called the step name, IEFBR14, or any name that will help you identify the step. In a large job, with many steps, unique step names can assist you when diagnosing problems. The choice is up to you.
- EXEC** Identifies a step job. It must be present.
- PGM=** Specifies the name of the program to be executed. In this case the program name is IEFBR14.

EXEC Statement Parameters



★ EXEC statement

▶ Region size

... REGION=4096K

▶ Process conditions from previous steps

... COND=(0,NE)

... COND=(0,NE,STEP1)

Figure 148. EXEC statement parameters

4.1.3.2 EXEC statement parameters

Two other parameters that you might use on the *EXEC* are:

- The *REGION* parameter specifies the quantity of virtual storage (or central storage when *ADDRSPC=REAL* coded) a step requires. If no *REGION* parameter is specified, the system uses an installation default specified at JES initialization. Some programs may need more storage than is allowed by default. To permit the program to get more storage, you can code the *REGION* parameter as follows:

```
//STEP1 EXEC PGM=progname,REGION=4096K.
```

This permits the programs to get up to 4 MB of storage below 16 MB and up to the installation default storage above 16 MB (IBM default is 32 MB).

- The *COND* parameter is used to inform the system to test return codes from previous job steps and determine whether to bypass this job step. You can specify one or more tests on the *COND* parameter, and you can test return codes from particular job steps or from every job step that has completed processing. If the test conditions are satisfied, the system evaluates the *COND* parameter as *true* and *bypasses* the job step. If the test conditions are not satisfied, the system evaluates the *COND* parameter as *false* and executes the job step. For example:

```
//STEP2 EXEC PGM=IEFBR14,COND=(0,NE)
```

With this statement, the system checks the return code from *all* previous steps, and if they were *not equal* to zero, then does not execute this step. You could also check for return codes that are *E*qual to, *G*T (greater than), *L*T (less than), *G*E (greater than or equal to), and *L*E (less than or equal to). You can check the return code in a specific step by coding:

```
//STEP2 EXEC PGM=IEFBR14,COND=(0,NE,STEP1)
```

With this statement, if STEP1 ends with return code zero, then STEP2 will be executed.

DD Statement Parameters: DSN



★ DD name

- ▶ As referred by the program

★ DSN

```
000100 //MYUSER01 JOB (ITSO),'IEFBR14',CLASS=A,MSGCLASS=X
000200 /*-----This is a comment line -----
000300 //STEP1 EXEC PGM=IEFBR14
000400 //NEWDD DD DSN=MYUSER.IEFBR14.TEST.NEWDD,
000500 // DISP=(NEW,CATLG,DELETE),
000600 // UNIT=SYSDA,
000700 // SPACE=(CYL,(10,10,45)),
000800 // LRECL=80,
000900 // BLKSIZE=3120
```

Figure 149. JCL DD statements: DDname, DSN

4.1.4 DD statement parameters

It should be stated at this time that the program IEFBR14 actually does nothing. But it enables you to submit a valid job and the system can process the DD statements identified in the JCL. This allows you to allocate, delete, catalog, and uncatalog data sets by using a batch process.

At job initialization, once the program IEFBR14 has been located, the system allocates the data sets specified in the DD statements, then the control is passed to program IEFBR14, that does nothing, and you get notified that your job is done.

Figure 150 shows the DD statement that identifies the data set we want to create.

```
000100 //MYUSER01 JOB (ITSO),'IEFBR14',CLASS=A,MSGCLASS=X
000200 /*-----This is a comment line -----
000300 //STEP1 EXEC PGM=IEFBR14
000400 //NEWDD DD DSN=MYUSER.IEFBR14.TEST.NEWDD,
000500 // DISP=(NEW,CATLG,DELETE),
000600 // UNIT=SYSDA,
000700 // SPACE=(CYL,(10,10,45)),
000800 // LRECL=80,
000900 // BLKSIZE=3120
```

Figure 150. The DD statement and DISP parameter

This statement shows some of the parameters that can be associated with a *DD* statement.

- *DDname* - Used to give a name to DD. *NEWDD* is the name assigned in the example. The programs refer to DD names instead of dsnames. So, unless a program allocates dynamically, all ddnames referred by a program must be coded. For example, if a program in a step needs a file identified as *OUTFILE*, you must code a DD named *OUTFILE*, identifying the relevant data set. In the example, *IEFBR14* does not use data sets. So you can choose any ddname you want.
- *DSN* - Used to identify the data set. In the example *MYUSER.IEFBR14.TEST.NEWDD* is the data set.

Note: You can direct outputs to JES by coding *SYSOUT* instead of *DSN*. In the following example, the ddname *SYSPRINT* is being directed to a *SYSOUT*, *CLASS=B*

```
//SYSPRINT DD SYSOUT=B
```

IEFBR14 does not put files, so you can use that resource in this case.

DD Statement Parameters: Disp, UNIT



★ DISP parameter

- ▶ Control concurrent access to data set
- ▶ What to do with the data set at end of processing

★ UNIT parameter

- ▶ What type of device is the data set allocated on

```
000100 //MYUSER01 JOB (ITSO),'IEFBR14',CLASS=A,MSGCLASS=X
000200 /*-----This is a comment line -----
000300 //STEP1 EXEC PGM=IEFBR14
000400 //NEWDD DD DSN=MYUSER.IEFBR14.TEST.NEWDD,
000500 // DISP=(NEW,CATLG,DELETE),
000600 // UNIT=SYSDA,
000700 // SPACE=(CYL,(10,10,45)),
000800 // LRECL=80,
000900 // BLKSIZE=3120
```

Figure 151. JCL DD statement parameters: Disp, Unit

4.1.4.1 DD statement: DISP parameter

The DISP parameter describes the status of a data set to the system and tells what to do with the data set after termination of the step or job. You specify this value for both normal and abnormal termination.

The first field identifies the *STATUS* of the data set and how to control access to it. It can be:

- NEW** Indicates the data set will be created and the job will have exclusive control of the data set. That means no other job can access this data set until the last step in this job that refers to this data set ends. NEW is the default.
- OLD** Indicates the data set exists and the job requires exclusive access to it.
- MOD** Indicates that if the data set exists, data will be appended to the end of the data set, otherwise, a new data set will be created. The job requires exclusive access to data set.
- SHR** Indicates that the data set can be shared by other users.

The second field in the DISP parameter indicates to the system what to do with the data set when the step finishes *NORMAL*. It can be:

- CATLG** Catalog the data set
- UNCATLG** Uncatalog the data set
- DELETE** Delete the data set

PASS Pass the data set to the subsequent steps

KEEP Keep the data set intact

The third field in the DISP parameter indicates the *ABNORMAL* completion action. It can be: DELETE, CATLG, UNCATLG, or KEEP.

In the example, the status field specifies to create the data set. In the normal termination of the step, to catalog. In abnormal to delete the data set. To delete a data set that exists you code its DSN and DISP=(OLD,DELETE,DELETE).

4.1.4.2 DD statement: UNIT parameter

This parameter identifies the device or type of device on which the data set will be allocated. You use this parameter to specify the number of devices to be used. If the data set exists, you only need to specify the device type if the data data is not cataloged. Most installations now administer disk storage with the Storage Management System (DFSMS/MVS). With SMS, you do not need to use the UNIT parameter to specify a device for SMS-controlled data sets. Some common examples are:

UNIT=SYSDA Allocates the data set on a direct access device (DASD)

UNIT=3390 Allocates the data set on a 3390 type disk.

UNIT=SYSALLDA Allocates the data set on a direct access device (DASD)

UNIT=TAPE Allocates the file on a TAPE device.

SPACE, LRECL, BLKSIZE



- ★ SPACE parameter
 - ▶ How much space to give the data set
- ★ LRECL parameter
 - ▶ What is the size of each record in the data set
- ★ BLKSIZE parameter
 - ▶ What is the block size (records in a block)

```
000100 //MYUSER01 JOB (ITSO),'IEFBR14',CLASS=A,MSGCLASS=X
000200 /*-----This is a comment line -----
000300 //STEP1 EXEC PGM=IEFBR14
000400 //SYSPRINT DD SYSOUT=*
000500 //NEWDD DD DSN=MYUSER.IEFBR14.TEST.NEWDD,
000600 // DISP=(NEW,CATLG,DELETE),
000700 // UNIT=SYSDA,
000800 // SPACE=(CYL,(10,10,45)),
000900 // LRECL=80,
001000 // BLKSIZE=3120
```

Figure 152. DD statement parameters: Space, Lrecl, blksize

4.1.4.3 DD statement: SPACE parameter

The *SPACE* DD parameter is required for allocating data sets on DASD. It identifies the space allocation required for your data set. Before a data set can be created on disk the system must know how much space the data set will require and how the space is to be measured. You can code it as follows:

SPACE=(type,(primary-qty,second-qty,directory))

Where *type* can be:

TRK Requests that space be allocated in tracks.

CYL Requests that space be allocated in cylinders.

block length You specify here the block length to request an allocation in number of blocks. Indicates that the values specified for primary and secondary allocations are block quantities, and directs the system to compute the number of tracks to allocate using a block length. For example, `SPACE=(3150,(5,1))` means that the system will allocate five blocks, each block having 3150 bytes, as primary space and one block as secondary space.

record length Specifies that the average record length in bytes will be used to allocate space. This is only applicable if SMS is active and the *AVGREC* parameter is coded.

The system allocates DASD space in whole tracks. The number of tracks required depends on how the records are blocked.

primary-qty specifies the initial allocation amount.

secondary-qty specifies an additional allocation amount. The system does not allocate additional space until it is needed.

directory you must code for a partitioned data set, to indicate the number of blocks the system must reserve for the directory. Partitioned Data Sets Extended (PDSE) grow dynamically, if you specify directory size, SMS uses the size you specify only if you later convert the PDSE to a PDS. You omit this parameter for sequential data sets.

Some examples of how you can code are:

SPACE=(TRK,4) To allocate space to a sequential data set, requesting four tracks, only primary space.

SPACE=(CYL,(5,2)) To allocate space to a sequential data set, five cylinders as primary allocation space and two cylinders as secondary.

SPACE=(CYL,(10,,100)) To allocate space to a partitioned data set, only primary allocation of 100 cylinders and 100 directory blocks.

4.1.4.4 LRECL and BLKSIZE parameters

Programs access data sets through ddnames. The ddnames are defined in the programs. Like them, data set characteristics like data set organization, logical record size and record size are also defined for the programs. During the allocation process you have to specify some of these characteristics. Some DD statement parameters you can use to specify the data set characteristics are:

LRECL Identifies the data set logical record size.

BLKSIZE Specifies the maximum length, in bytes, of a block.

RECFM Identifies the the record format.

DSORG Identifies the data set organization. If you do not specify DSORG, the system uses the information in SPACE to determine if the data set should be sequential or partitioned

These parameters are part of Data set Control Block (DCB) information and can be coded in the JCL as follows:

```
// DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
```

or as individual parameters, without the need to specify the DCB=(...) parameter, for example:

```
// LRECL=80,  
// BLKSIZE=3120,  
// RECFM=FB,  
// DSORG=PO
```

If BLKSIZE is not specified, the system will determine what it considers to be an optimum block size for the device type on which the data set will be allocated. For a data set with fixed record format (all records with same size), the BLKSIZE must be a multiple of LRECL. For variable record size, the BLKSIZE must be a multiple of the greatest record plus four.

Submitting a Job



```
Command ==> SUB                               Scroll ==> CSR
***** Top of Data *****
000100 //MYUSER01 JOB (ITSO),'IEFBR14',CLASS=A,MSGCLASS=X
000200 //*-----This is a comment line -----
000300 //STEP1   EXEC PGM=IEFBR14
000400 //NEWDD   DD DSN=MYUSER.IEFBR14.TEST.NEWDD,
000500 //          DISP=(NEW,CATLG,DELETE),
000600 //          UNIT=SYSDA,
000700 //          SPACE=(CYL,(10,10,45)),
000800 //          LRECL=80,
000900 //          BLKSIZE=3120

JOB MYUSER01(jobnumber) SUBMITTED
***
```

Figure 153. Submitting a job

4.1.5 Submitting a job

You have now created data set MYUSER.PRIVATE.JCLLIB, built your JCL member, IEFBR14, and you are ready to *SUBMIT* this job.

To submit the JCL stream, Enter SUB (submit) on the command line. The TSO/E command processor sends the JCL statements to JES.

After entering the command, you should receive the following message indicating that your job was submitted successfully:

```
JOB jobname(jobnumber) SUBMITTED
***
```

ISPF Option 3.4 Panel



```

                                Data Set List Utility

Option ==>

    blank Display data set list          P Print data set list
      V Display VTOC information          PV Print VTOC information

Enter one or both of the parameters below:
Dsname Level . . . MYUSER
Volume serial . .

Data set list options
Initial View . . . 1  1. Volume          Enter "/" to select option
                   2. Space             / Confirm Data Set Delete
                   3. Attrib            / Confirm Member Delete
                   4. Total

When the data set list is displayed, enter either:
"/" on the data set list command field for the command prompt pop-up,
an ISPF line command, the name of a TSO command, CLIST, or REXX exec,
"=" to execute the previous command.
```

Figure 154. ISPF option 3.4

4.1.5.1 ISPF option 3.4

If you select ISPF option 3.4 you will access the Data Set List Utility panel as shown in the visual.

You can use this option to manage all data sets that you have access to. Move the cursor to the *Dsname Level* field and enter the high level qualifier of the data set(s) you want to work with. This example uses the *MYUSER* HLQ.

When you press the Enter key, the data set list is displayed, as shown in the next visual.

Data Set List for MYUSER



```
DSLISL - Data Sets Matching MYUSER                               Row 1 of 5
Command ===>                                                    Scroll ===> CSR

Command - Enter "/" to select action      Message              Volume
-----
          MYUSER                                           *ALIAS
          MYUSER.BROADCAST                               SYS001
%          MYUSER.PRIVATE.JCLLIB                          SYS002
          MYUSER.ISPF42.ISPPROF                          SYS001
          MYUSER.SPFLOG1.LIST                             SYS001
          MYUSER.SPFTEMP0.CNTL                            SYS001
***** End of Data Set list *****
```

Figure 155. Data set list for MYUSER

4.1.5.2 Data set list from ISPF option 3.4

The visual shows the data sets that have the high level qualifier, MYUSER.

You can select data sets for processing by entering any of the line commands shown in Figure 156 to the left of the data set name. As you can see, this is a very comprehensive list of commands to enable you to manage you data sets. To learn about these commands, type HELP in the COMMAND field.

```
V - View data set      RA - Refadd to Reflist    Z - Compress data set
B - Browse data set   C - Catalog data set      F - Free unused space
E - Edit data set     U - Uncatalog data set   = - Repeat last comma
D - Delete data set   P - Print data set        CO - Copy data set
R - Rename data set   PX - Print index listing  MO - Move data set
I - Data set information M - Display member list RS - Reset statistics
S - Information (short) X - Exclude data set   NX - Unexclude data se
TSO commands, CLIST, or REXX exec
```

Figure 156. Displayed data set list commands

Typing *E* next to data set MYUSER.PRIVATE.JCLLIB will display a list of the members contained within that data set. Figure 159 on page 227 shows an example of a member list. At this stage you only have one member, IERBR14, but over time this will grow to include members that assist you with all the functions you perform as a systems programmer.

```

EDIT          MYUSER.PRIVATE.JCLLIB          Row 00001 of 00001
Command ==>          Scroll ==> CSR
      Name      Prompt      VV MM      Changed      Size  Init  Mod
----- IEFBR14          01.03 99/04/21 16:04      10    3    7
      **End**

```

Figure 157. Data set member list

Tab down to the IEFBR14 member and press Enter which will allow you to *edit* the member. The default action on the member will be based on the action you selected for the data set. For example, if you selected *Edit* for the data set, the default action for the member selected will be edit. You can issue the *Select* command next to the member but this is not necessary.

Entering a / at Data Set



```

                                Data Set List Actions

Data Set: MYUSER.PRIVATE.JCLLIB

DSLIST Action
  1. Edit                      12. Compress
  2. View                      13. Free
  3. Browse                    14. Print Index
  4. Member List              15. Reset
  5. Delete                   16. Move
  6. Rename                   17. Copy
  7. Info                     18. Refadd
  8. Short Info              19. Exclude
  9. Print                    20. Unexclude 'NX'
 10. Catalog                  21. Unexclude first 'NXF'
 11. Uncatalog                22. Unexclude last 'NXL'

Select a choice and press ENTER to process data set action.
F1=Help      F2=Split      F3=Exit      F7=Backward
F8=Forward   F9=Swap        F12=Cancel

```

Figure 158. Action option list

4.1.5.3 Data set list from ISPF option 3.4

Typing a / on the MYUSER line causes ISPF to display a panel with all the list actions you can use in that column. You can choose the option number or, by going back to the member list, type the command on the COMMAND line.

Figure 159 shows how to select a member in a PDS by typing its name directly in the DSLIST panel, IEFBR14.

```

DSLIST - Data Sets Matching MYUSER                      Row 1 of 5
Command ==>>>                                         Scroll ==>> CSR

Command - Enter "/" to select action      Message      Volume
-----
      MYUSER                                *ALIAS
      MYUSER.BROADCAST                      SYS001
E     MYUSER.PRIVATE.JCLLIB(IEFBR14)        SYS002
      MYUSER.ISPF42.ISPPROF                 SYS001
      MYUSER.SPFL0G1.LIST                   SYS001
      MYUSER.SPFTempo.CNTL                  SYS001
***** End of Data Set list *****

```

Figure 159. Creating a PDS member using option 3.4

You can do that even if the data set is empty.

Spool Display and Search Facility - SDSF

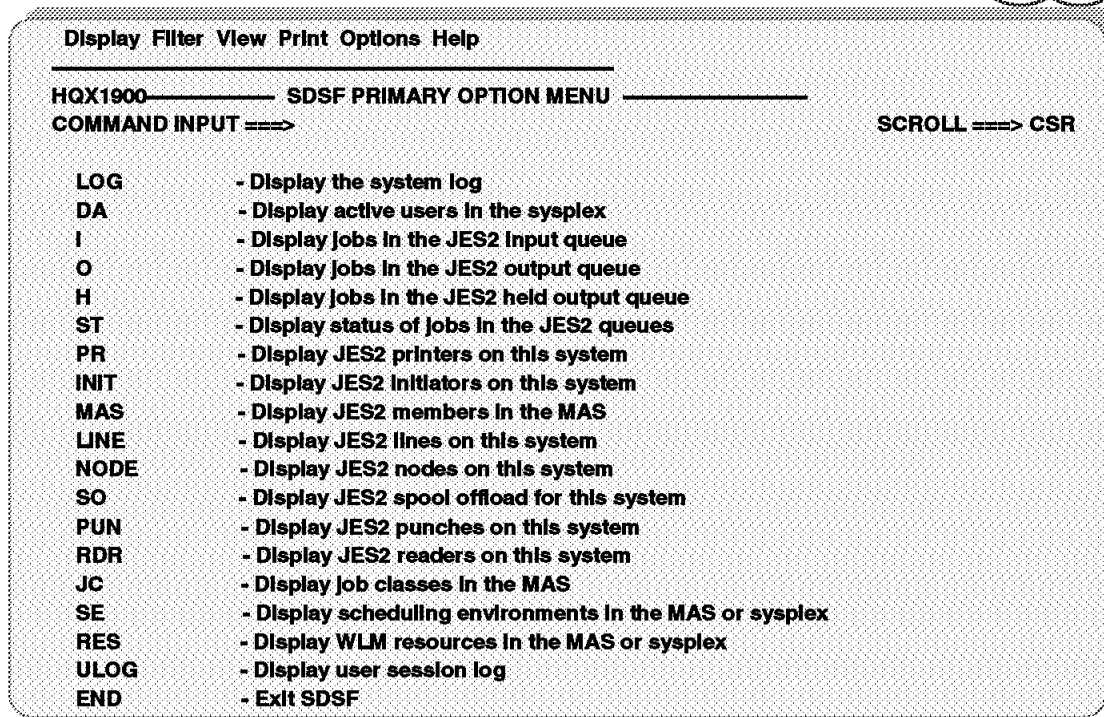


Figure 160. SDSF Primary Option menu

4.1.6 Introduction to the Spool Display and Search Facility (SDSF)

During this section we will discuss the SDSF facility and how you can use its functions to monitor and manage your workloads. We will see how SDSF can be used to display job input and output data, and purge (delete) jobs that are on the input, output, or held queues. We will review the monitoring functions of SDSF that enable you to evaluate the current workload, and will enable you to cancel, hold, or reschedule work.

SDSF can be invoked from ISPF menus, but the setting of the options is often customized by each site differently. You will have to review your site's ISPF menus to find the SDSF option. Alternatively, issuing the *TSO SDSF* command, from the *Command ==>* line will invoke SDSF. After choosing this option, the panel you will receive may not have the same options as shown in Figure 160. The options vary according to the security level of the user. The authority to perform functions within these options will also vary according to the security level of the user. It is possible to control most system functions by using the SDSF facility. The scope of the functions range from reviewing job output, controlling the processing of jobs, both their input and output, printer control, operator functions, and system task administrative functions.

The *ST* option shows all jobs in the JES2 queues: input, held output, and output. Before choosing any option, you place the cursor in the menu action bar, at *Options* and press the Enter key. If option 5 shows *Set display values to ON*, then choose it. With this option *ON*, the selection criteria are shown in the panel. You get the same result by issuing the SDSF command *SET DISPLAY ON* in the *COMMAND* line.

SDSF - Viewing the JES2 output files

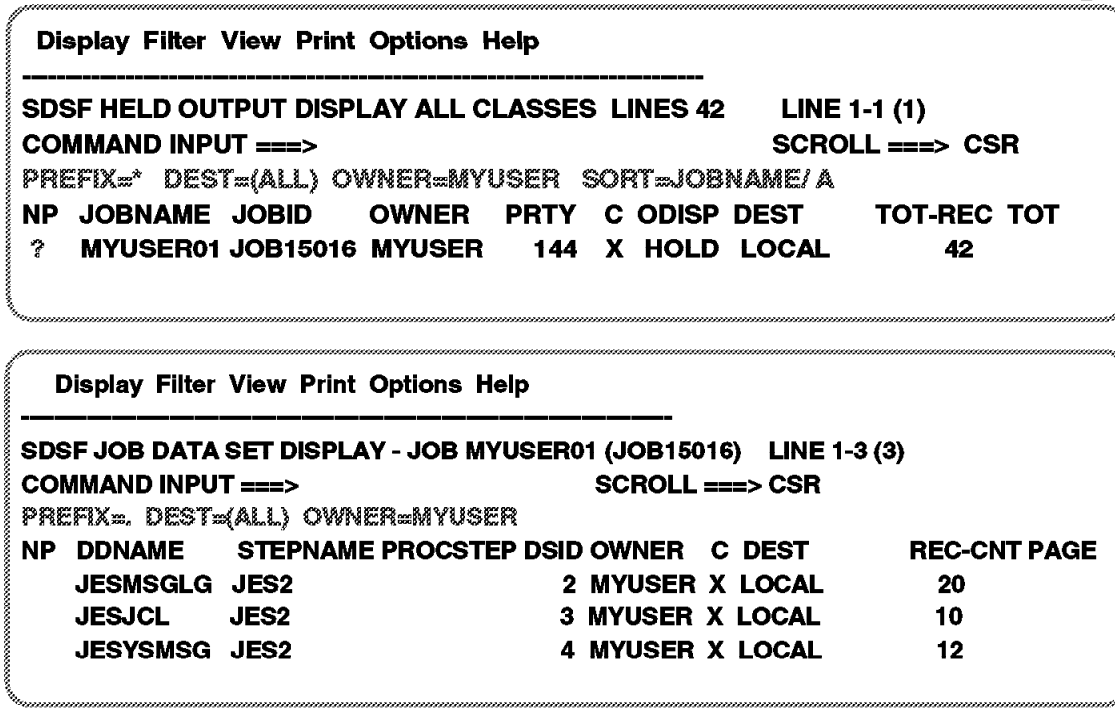


Figure 161. SDSF - Viewing the JES2 output files

4.1.6.1 Viewing the job execution output

Typing *H* in the Menu Panel, allows you to see the output that was created during the execution of your batch job. It is saved on the JES spool data set, unless you requested the output to be automatically purged by setting a MSGCLASS that has been defined to not save output. Depending on the MSGCLASS you chose, it could be in the *Output queue*; in this case you should choose the *O* option.

The first screen shown in Figure 161 displays a list of the jobs we have submitted and whose output we have directed to the *HELD* (Class X) queue, as identified in the MSGCLASS=X parameter in the job card. In our case only one job has been submitted and executed. Therefore, we only have one job on the held queue.

Issuing an *?* command in the NP column will display the output files generated by job 15016. The second screen shown in Figure 161 displays three ddnames: JES2 messages log file, JES2 JCL file, and JES2 system messages file. This option is useful when you are seeing jobs with many files directed to SYSOUT and you want to display one associated with a specific step. You issue an *S* in the NP column to select a file you want. To see all files, instead an *?* type *S* in the NP column.

J E S 2 J O B L O G -- S Y S T E M S C 4 2 --

```

11.12.31 JOB15016 ---- FRIDAY,    23 APR 1999 ----
11.12.31 JOB15016 IRR010I  USERID MYUSER  IS ASSIGNED TO THIS JOB.
11.12.31 JOB15016 ICH70001I MYUSER  LAST ACCESS AT 08:25:43 ON FRIDAY,
11.12.31 JOB15016 $HASP373 MYUSER01 STARTED - INIT A    - CLASS A - SYS
11.12.31 JOB15016 IEF403I MYUSER01 - STARTED - TIME=11.12.31
11.12.31 JOB15016 -                               --TIMINGS (
11.12.31 JOB15016 -JOBNAME  STEPNAME  PROCSTEP   RC    EXCP  CPU    SR
11.12.31 JOB15016 -MYUSER01      STEP1      00     12  .00
11.12.31 JOB15016 IEF404I MYUSER01 - ENDED - TIME=11.12.31
11.12.31 JOB15016 -MYUSER01 ENDED.  NAME-IEFBR14                TOTAL CPU
11.12.31 JOB15016 $HASP395 MYUSER01 ENDED

```

----- JES2 JOB STATISTICS -----

23 APR 1999 JOB EXECUTION DATE

10 CARDS READ

42 SYSOUT PRINT RECORDS

0 SYSOUT PUNCH RECORDS

3 SYSOUT SPOOL KBYTES

0.00 MINUTES EXECUTION TIME

1 //MYUSER01 JOB (ITSO),' IEFBR14', CLASS=A,MSGCLASS=X

/**

2 //STEP1 EXEC PGM=IEFBR14

3 //SYSPRINT DD SYSOUT=*

4 //NEWDD DD DSN=MYUSER.IEFBR14.TEST.NEWDD,

// DISP=(NEW,CATLG,DELETE),

// UNIT=SYSDA,

// SPACE=(CYL,(10,10,45)),

// LRECL=80,

// BLKSIZE=3120

ICH70001I MYUSER LAST ACCESS AT 08:25:43 ON FRIDAY, APRIL 23, 1999

IEF236I ALLOC. FOR MYUSER01 STEP1

IEF237I JES2 ALLOCATED TO SYSPRINT

IGD100I 256C ALLOCATED TO DDNAME NEWDD DATACLAS ()

IEF142I MYUSER01 STEP1 - STEP WAS EXECUTED - **COND CODE 0000**

IEF285I MYUSER.MYUSER01.JOB15016.D0000101.? SYSOUT

IEF285I **MYUSER.IEFBR14.TEST.NEWDD CATALOGED**

IEF285I **VOL SER NOS= TOTTSJ.**

IEF373I STEP/STEP1 /START 1999113.1112

IEF374I STEP/STEP1 /STOP 1999113.1112 CPU OMIN 00.00SEC SRB

IEF375I JOB/MYUSER01/START 1999113.1112

IEF376I JOB/MYUSER01/STOP 1999113.1112 CPU OMIN 00.00SEC SRB

Figure 162. SDSF displaying the output job

Figure 162 shows the output for our job. The most important things to note are:

1. The *RC* or Return Code value is *00* which indicates a successful completion of the step.
2. The *COND CODE* or Condition Code is *0000* which indicates a successful completion of the job.
3. The messages in Figure 163 on page 232 show that the data set MYUSER.IEFBR14.TEST.NEWDD was allocated on disk volume TOTTSJ, and was cataloged.

```
IEF285I: MYUSER.IEFBR14.TEST.NEWD          CATALOGED
IEF285I: VOL SER NOS= TOTTSJ.
```

Figure 163. JES2 system messages

SDSF - Display Active Users (DA)



```
Display Filter View Print Options Help
-----
SDSF DA SC67 SC67 PAG 0 SIO 7 CPU 6/ 7 LINE 1-25 (64)
COMMAND INPUT ==> SCROLL ==> PAG
PREFIX=* DEST=LOCAL OWNER=* SORT=JOBNAME/A
NP JOBNAME STEPNAME PROCSTEP JOBID OWNER C POS DP REAL PAGING SIO
 *MASTER* STC06373 +MASTER+ NS FF 1369 0.00 0.00
ALLOCAS ALLOCAS NS FF 190 0.00 0.00
ANTAS000 ANTAS000 IEFPROC NS FE 1216 0.00 0.00
ANTMAIN ANTMAIN IEFPROC NS FF 4541 0.00 0.00
APPC APPC APPC NS FE 2653 0.00 0.00
ASCH ASCH ASCH NS FE 267 0.00 0.00
BPXOINIT BPXOINIT BPXOINIT LO FF 315 0.00 0.00
CATALOG CATALOG IEFPROC NS FF 1246 0.00 0.00
CICSPAAY CICSPAAY CICS520 STC06504 STC NS FE 4330 0.00 0.00
CONSOLE CONSOLE NS FF 597 0.00 0.00
DFRMM DFRMM IEFPROC STC06363 STC NS FE 510 0.00 0.00
DFSMSHSM HSMSC67 DFSMSHSM STC13178 STC NS FE 6199 0.00 0.00
DUMPSRV DUMPSRV DUMPSRV NS FF 160 0.00 0.00
FTPDVMS1 STEP1 STC06477 STC LO FF 470 0.00 0.00
FTPDOE1 STEP1 STC06475 FTPDOE LO FF 469 0.00 0.00
GRS GRS NS FF 894 0.00 0.00
IEFSCHAS IEFSCHAS NS FF 25 0.00 0.00
IMWEBSUF IMWEBSUF WEBSRV STC15245 WEBSRV IN FE 15T 0.00 0.00
```

Figure 164. SDSF - display active users (DA)

4.1.6.2 Displaying the active tasks

SDSF provides the ability to monitor the current system workload. The DA command displays the active tasks and provides information about each task. This information includes CPU usage for each task, the amount of CPU time that a task has used, and the Input/Output related EXCP statistics. The visual displays some of the data that this facility captures. Press *PF11*, to go right and see all the available fields.

You can customize your SDSF panels by choosing the *View* option in the action bar and then the *Arrange* option. You do that for each panel where *Arrange* is available and choose the the fields in order. SDSF stores the information in your profile data set and uses them the next time you enter any SDSF options.

4.1.6.3 Issuing MVS and JES commands under SDSF

On the *COMMAND INPUT ==>*line of any SDSF panel you can issue any MVS or JES2 command following a slash (/) if you are authorized. If the command is too long, type *+* and two more lines will appear to allow you to type the rest of the command. If you have authority, you can use the *ULOG* option to see only your commands and their response.

SDSF provides a sort of Actions commands you can use in the *NP* column. Some of them you may not have authority to issue. In this case, SDSF issues a message in the top right of the panel. When you choose an action command, SDSF issues the system command that corresponds to the action you chose.

To display which action commands can be used in an SDSF panel, issue the HELP command in each option panel. Then choose Option 3 - *Action characters*:. A panel listing all the action commands you can use in that option appears.

Proceed with caution in the SDSF DA panel

If you issue a C (Cancel) action command in the DA display it will cancel any task you select if you have the authority. Use this command with caution if you are displaying major production tasks. Commands issued in the DA display are issued against running tasks. Incorrect or careless use can cause major problems.

4.1.6.4 Viewing the output queue

The SDSF 0 command will display the non-held output queue. If you have the authority to review all output you might want to use the PREFIX command to limit the entries in the display. For example,

- *PREFIX*, with no parameter will display all jobs for which you are authorized.
- *PREFIX MQ** will display all jobs that start with the name MQ.
- *PREFIX MYUSER** will display all jobs that begin with the name MYUSER.

The default, when you first enter SDSF is your logon ID prefix. If SDSF is not displaying what you expect, issue the SET DISPLAY ON command. This controls the display of values you have set for PREFIX, DEST, OWNER, SORT, and FILTER.

Figure 165 shows an output display using prefix mq*.

| NP | JOBNAME | JOBID | OWNER | PRTY | C | FORMS | DEST | TOT-REC |
|----|----------|----------|---------|------|---|-------|-------|---------|
| | MQWFS11 | STC11032 | HKOLATA | 96 | A | STD | LOCAL | 5,328 |
| | MQWFS11 | STC11102 | HKOLATA | 96 | A | STD | LOCAL | 6,222 |
| | MQWFS11 | STC11108 | HKOLATA | 96 | A | STD | LOCAL | 7,227 |
| | MQWFS11 | STC11114 | HKOLATA | 96 | A | STD | LOCAL | 3,780 |
| | MQWFADM1 | STC14544 | SRVRID1 | 96 | A | STD | LOCAL | 18,771 |

Figure 165. SDSF output display

Chapter 5. MVS delivery and installation

This chapter describes the OS/390 delivery options and the download process using the ServerPac option. Chapter 4 explains the basics of the OS/390 implementation.

Once positioned on OS/390, you can receive function in new levels approximately every six months. This predictable release cycle should reduce planning time. Because each new level is comprehensively tested, the quality of the operating system is improved. Once your initial migration is complete, you can expect simplified ordering, planning, and installing, freeing you to increase the value of your computing environment to your business and deliver better service to your end users.

OS/390 consists of base elements and optional features. The base elements (or simply elements) deliver essential operating system functions. The base elements consist of former MVS products and new functions. The base elements are listed in Volume 1. When you order OS/390, you receive all of the base elements.

The optional features (or simply features) are orderable with OS/390 and provide additional operating system functions. Like the base elements, the optional features consist of former MVS products and new functions. The optional features are listed in Volume 1.

A new release of OS/390 is available every six months. At the end of every March and September, a new release will become generally available, with shipments starting at the beginning of April and October. Note that not every element and feature in a given release will contain new function. However, those elements and features that don't receive new function will have all eligible service included.

Because the base elements and optional features of OS/390 are integrated into a single package with compatible service levels, you must install, with few exceptions, the entire OS/390 product. You can install OS/390 using one of several IBM packages and they are discussed in this chapter.

OS/390 - Installation overview

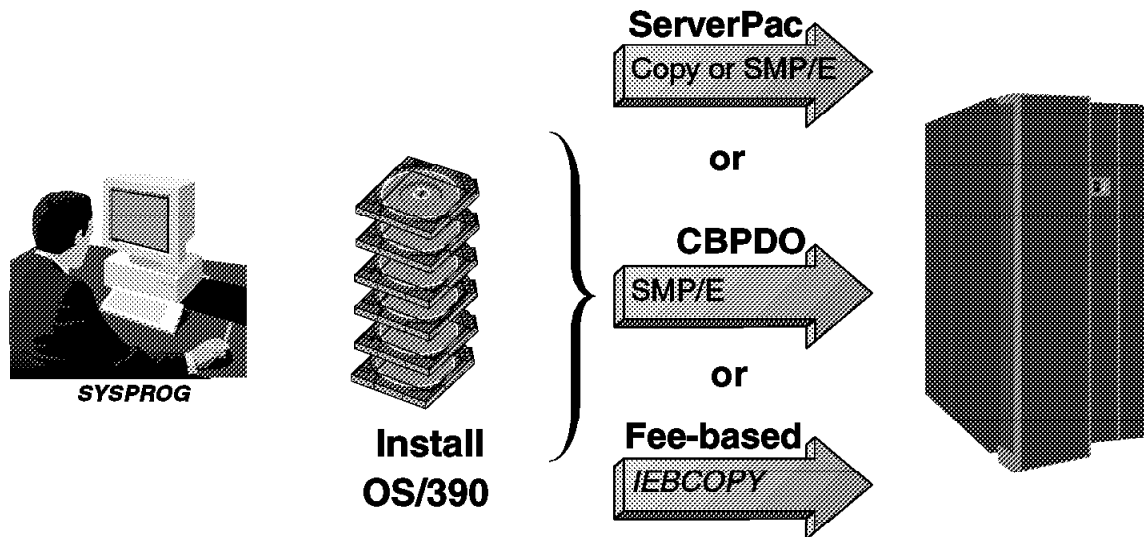


Figure 166. Installation overview

5.1 Introduction to OS/390

Prior to OS/390, your large applications ran on an MVS operating system that consisted of the Basic Control Program (BCP), the Data Facility Product (DFSMSdfp), and Job Entry Subsystem (JES2 or JES3), plus a collection of other software products that the applications required, such as Interactive System Productivity Facility (ISPF) and Time Sharing Option/Extensions (TSO/E), and so forth.

Recent functional additions might have included LAN services, distributed computing software, and application-enabling packages. You traditionally ran these products at various release levels, using a mix and match approach.

With the introduction of OS/390, all these products were integrated into a single product. You will no longer order new levels of some products but not of others; instead, you will order and install an entire set of products integrated into one functionally rich operating system.

5.1.1 Advantages of OS/390

The release cycle of OS/390 is approximately every six months, hence you expect to receive function in new levels from every release. This predictable release cycle should reduce your planning time.

Before the new release of OS/390 is shipped to you, each new function levels are comprehensively tested, hence the quality of the operating system is improved.

Once your initial migration to OS/390 is complete, you can expect simplified ordering, planning, and installing of the next OS/390 release.

5.1.2 OS/390 delivery options

You can install OS/390 using one of several IBM packages. Two of these packages are available at no additional charge when you license OS/390:

ServerPac This software delivery package consists of products and service for which IBM has performed the SMP/E installation steps and some of the post-SMP/E installation steps. You use the CustomPac Installation Dialog to install your system and complete the installation of the software it includes.

CBPDO The Custom Built Product Delivery Option (CBPDO) is a delivery package that consists of uninstalled products and unintegrated service. You must use SMP/E to install the individual OS/390 elements and features, and their service, before you can IPL.

Fee-based options are available as follows:

SystemPac SystemPac tailors OS/390 to your environment (such as DASD layout, migration of MSVCP/IOCP to IODF, and naming conventions) based on information provided to IBM. With this offering, selected non-IBM products can be integrated.

SoftwareXcel SoftwareXcel Installation Express (SIE) is available in the U.S. only, and provides pre-built OS/390 system packages in full volume dump format, tailored to customer hardware and software configurations. SIE includes on-site planning, installation, and package testing.

Entry Server Offering The Entry Server Offering, only available in selected countries, is a packaged solution that includes hardware, software, installation services, maintenance, and financing to help customers get current technology.

Note: When you order any one of the installation packages, you will receive a comprehensive installation guide, detailing the installation task step by step from the beginning of the installation until you IPL your system. For example, if you choose the ServerPac installation package, you receive the *ServerPac: Installing Your Order* tailored to your order for installation. This document is unique to your environment based on what you have ordered. In this chapter we will briefly discuss the installation steps using ServerPac.

5.1.3 ServerPac service level

For ServerPac orders, service is integrated with product code according to the following time-line:

- All the products in the ServerPac will incorporate the PTFs contained in the PUT packages that were available approximately two months before the OS/390 release became generally available.
- All products will incorporate HIPER and PTF-in-error(PE) fixes that are available approximately one week before your order is received.
- All OS/390 elements and features will incorporate additional service that has been through OS/390 integration testing.
- Your ServerPac order will also contain a service tape containing all unintegrated service that was available at the time your order was processed.

5.1.4 CBPDO service level

Remember, the service in the CBPDO orders is not integrated. You must receive and apply the service during the installation process. The service that is shipped with every CBPDO order is as follows:

- Service for all products previously ordered, as reflected in the customer profile for the customer number used when the order was placed. You can specify the starting date for these PTFs at order time. The default starting date is the last time that customer number was used to order a CBPDO (product or service) for that system release identifier (SREL).
- Service for all products, elements, and features that you have ordered. This includes all PTFs that became available between product general availability and the time of your order that have not been incorporated in the product FMIDs.

There is a *Memo to Users Extension* that comes with CBPDO describing the source IDs for service delivered on the CBPDO tape.

System and Installation Requirements

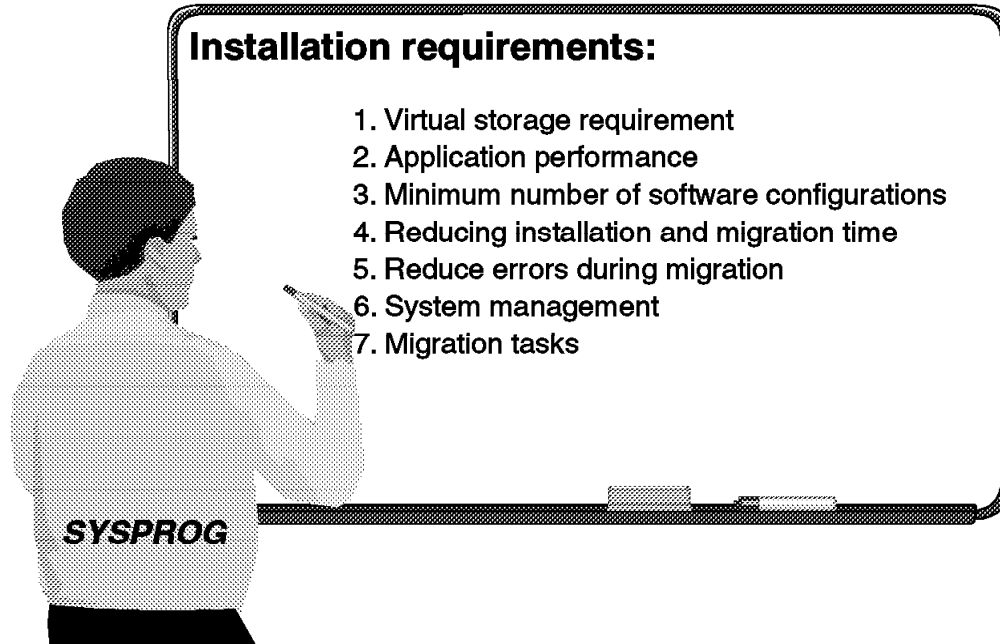


Figure 167. System and installation requirements

5.1.5 System and installation requirements

Having an installation plan helps you plan to make sure the software is able to meet your installation's requirement for software function. However, there are other things you should consider when planning to build a system. These additional requirements include:

- Virtual storage mapping
- Application performance
- Building a minimum number of system software configurations
- Reducing installation and migration time
- Reducing the opportunities for error during migration
- Making it easier to manage the system after it is in production
- Minimizing migration actions for the people who use the system

How you choose to meet all these requirements can have a significant effect on how much work is required to perform the tasks associated with each stage. Keep all these additional requirements in mind when you are planning to build a new system.

Reviewing Your Current System

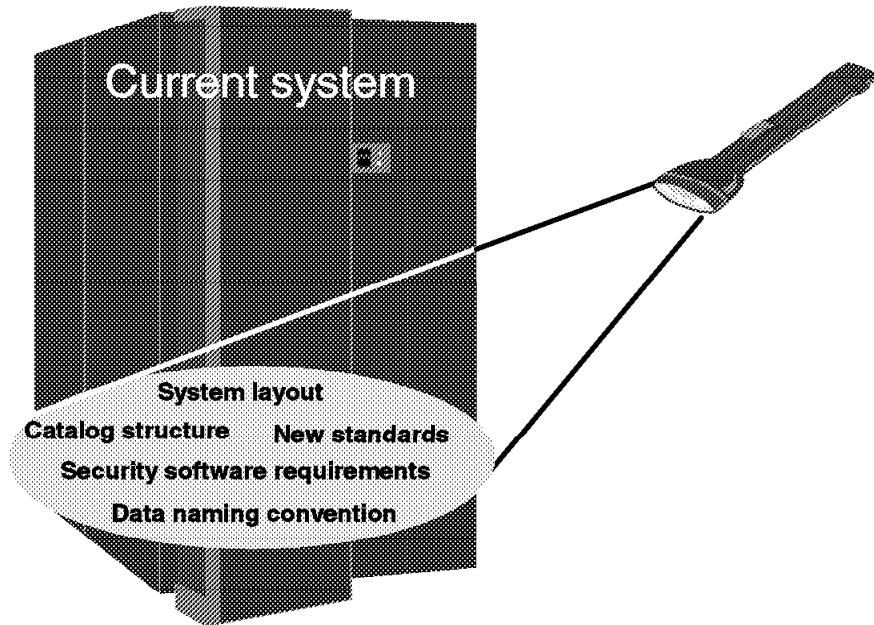


Figure 168. Reviewing your current system

5.1.6 Reviewing your current system

More often than not, you are planning to migrate from your current MVS system to the new release of OS/390. It is therefore very important to review the setup of your current environment while planning for the new system. Some of the things you should consider are:

- The system layout
- The catalog structure
- Data set naming conventions in your present environment
- Security software considerations
- New standards, if necessary

Depending on your order, the system target and distribution libraries may exceed more than one DASD volume, for example IBM 3390-3. You should define your new system layout to be prepared for future installation and easy cloning of your system. You can make use of the worksheets included in *IBM ServerPac for OS/390 Using the Installation Dialog*, SC28-1244, and define where the following new data sets should reside:

- Target data sets
- Distribution libraries data sets
- Master catalog and user catalogs
- Dialog and order data sets

The Driving and Target System

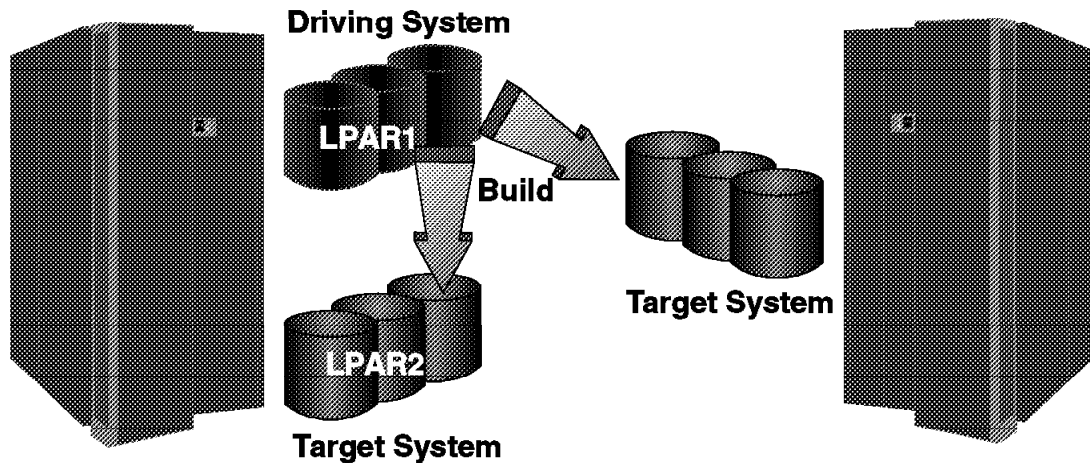


Figure 169. The driving and target system

5.1.7 The driving system and the target system

The *driving system* is the system image (both the hardware and software) that you use to install your new system image, the *target system*. You log on to the driving system and run jobs there to create or update the target system. Once the target system is built, it can be IPLed on the same hardware (same LPAR or different LPAR on the same processor) or different hardware than that used for the driving system.

To prepare the driving system before building the target system, you need to perform the following tasks:

- Identify the software requirement for the driving system using ServerPac or CBPDO.
- Identify the hardware requirement for the driving system.

You are also required to do some preparations for the target system:

- Choose the software products to install and identify requisities.
- Order OS/390 and related IBM products.
- Identify the hardware requirement for the target system.
- Identify the service needed for the target system.
- Decide whether to use the existing JES2 or JES3 with the new OS/390 release.

For more information on each of the requirements for both the driving and target system, see *OS/390 Planning for Installation*, GC28-1726, for the OS/390 release that you are installing.

OS/390 Installation using ServerPac

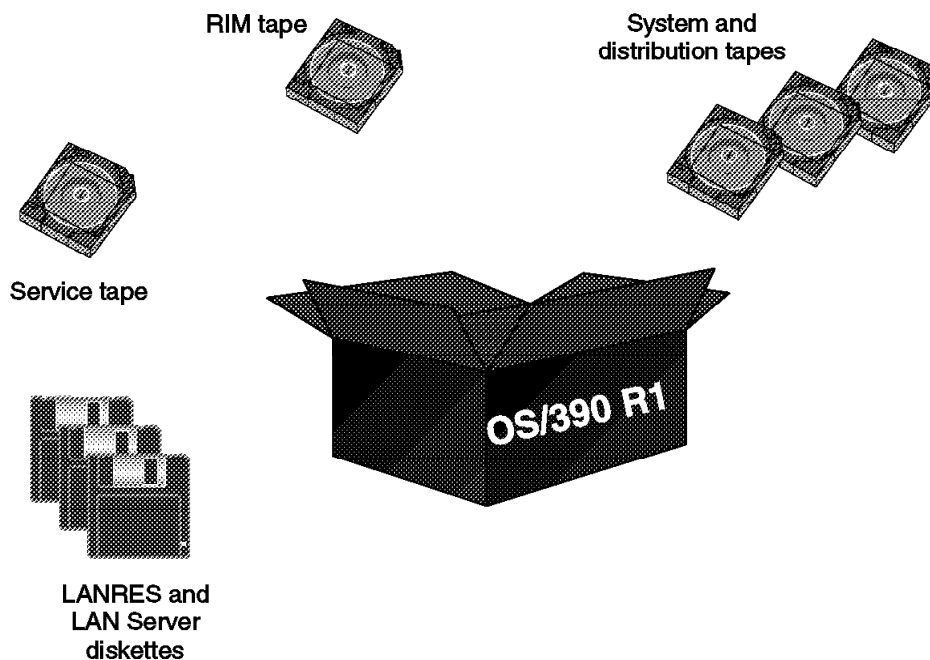


Figure 170. OS/390 installation

5.2 Installing OS/390 using ServerPac

This section describes the installation steps which are provided by *IBM ServerPac for OS/390 Using the Installation Dialog*, SC28-1244. Your OS/390 ServerPac order contains an ISPF dialog that you use to install OS/390. This dialog is called the CustomPac Installation Dialog because it is used to install all of IBM's CustomPac offering.

Before you begin the installation, you should:

- Review the contents of the ServerPac shipment that you received from IBM by checking the packing slip to make sure that you have a complete set of installation tapes and documentation.
- Make sure your user ID has ALTER authority for the following high-level qualifiers:
 - CPAC
 - SYS1
 - All product-specific high-level qualifiers for products that come with your package. You can find a listing of all qualifiers by using the A-ALIAS option of the dialog, or refer to Product Information in the Appendix of the document *ServerPac: Installing Your Order*.

5.2.1 Load RIM tape

The first step in installing OS/390 is to install the CustomPac dialogs from the RIM tape. Once they are installed, the dialogs do not have to be reinstalled with every order. They may be updated when initiated by IBM whenever you get a new order. Version checking invokes the update of the dialogs during the CustomPac RECEIVE function.

Throughout the installation of the dialogs, you are requested to define a CustomPac qualifier or the HLQ of your master data sets. Since the dialogs are permanently installed at your installation, you should not specify the IBM-supplied order number in the CustomPac qualifier.

The CustomPac Dialogs

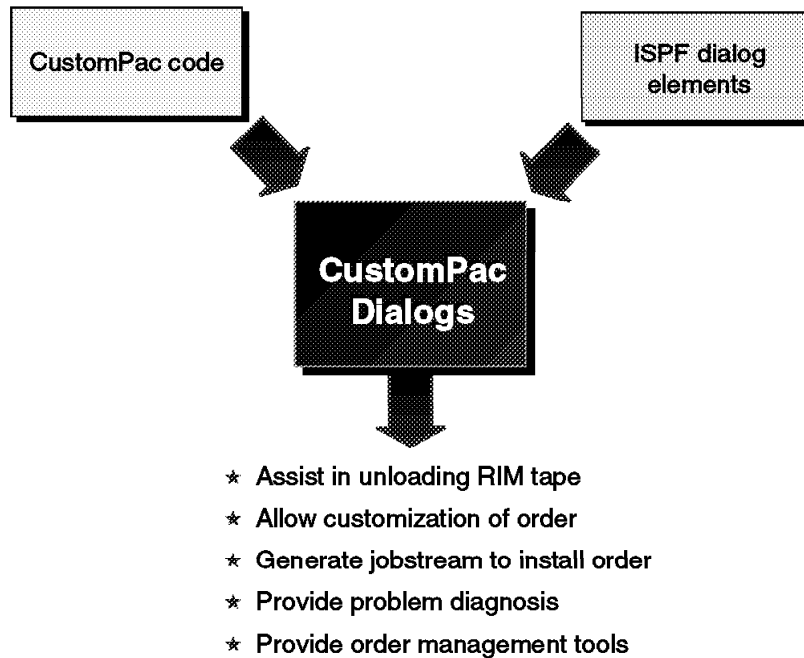


Figure 171. The CustomPac dialogs

5.2.2 Installing the CustomPac dialogs

The RIM tape contains the following sample procedures, JCL, jobs, and CLISTs:

| Name | Description |
|-----------------|--|
| LOADRIM | LOADRIM is the JCL to unload files from tape and the setup of the installation dialog. When you edit the LOADRIM sample JCL, you can choose the name of the master data sets, the unit name of your tape drives, and the VOLSER of the DASD which receives the installation dialog's data sets. Note: For the master and order data sets, you should use different HLQs. |
| SETUP | This is a sample LOGON procedure which includes the CustomPac dialog ISPF libraries. |
| CPPCSAMP | This sample CLIST can be used to set up the environment instead of modifying the LOGON procedure. CPPCSAMP uses LIBDEFs and should be the preferred method to allocate the CustomPac libraries and start the dialog. CPPCSAMP can be used after invoking ISPF. |
| CPPINIT | With the CPPINIT CLIST, you can set up the environment from native TSO. |
| PRTDOC | This sample job prints the CustomPac Installation dialog reference manuals. |

Note: HELP (PF1) is available on any panel. The HELP key is a very useful online help facility that explains every panel function in detail. Some panels have PRIM and LINE commands available. Using the HELP key allows you to get a description and example on how to use the commands.

Receiving the ServerPac Order



```
CustomPac ----- (C) IBM Corporation 1990-1996 -----  
OPTION ==> R
```

```
CustomPac INSTALLATION - Select The Option Of Your Choice
```

```
    The HLQ of your MASTER data set is : SYSPOK.MASTER
```

```
    R RECEIVE    - Receive an Order
```

```
    I INSTALL    - Install Orders
```

```
    ==> OS261466 (Order Nbr OR blank for ALL NEW Orders)
```

```
    O ORDER      - Order Information
```

Figure 172. Receiving the ServerPac order

5.2.3 Receiving the ServerPac order

Invoke the CustomPac CLIST *CPPCSAMP* to start the dialog. Receiving the order means you will copy the order from tape to DASD.

R receives the order. This unloads the control tables and installation jobs from the shipment tapes to your DASD. Selecting option *R* selects the Order Receive panel.

Order Receive Panel



```
CustomPac ----- Order Receive -----  
COMMAND ==>  
  
ORDER DETAILS  
  
Order Number ==> OS261466  
  
TAPE VolSer ==> R1466A TAPE Unit ==> 3490  
  
Order HLQ ==> SYSPOK.OS261466  
  
DASD VolSer ==> TOTTS2 DASD Unit ==> SYSDA  
  
Do You Want To Use VB Clists ==> N  
  
Edit JOB Stream Before Submitting ==> N
```

Figure 173. Order Receiving panel

5.2.4 Order Receive panel

After completion of the Receive option, a batch job is generated and submitted to download the order installation libraries from the shipment tape to DASD.

Panel details

Order Number This is your specific IBM-supplied order number, which is listed on the cover of the order documentation.

TAPE VolSer This is the volume serial number of the RIM tape.

TAPE Unit This is the unit type of your tape drives.

Order HLQ This is the HLQ used to allocate the order installation data sets. We recommend you include the order number as part of the qualifier.

DASD VolSer This is the VOLSER of the DASD where your order data sets are to be restored.

DASD Unit This is the unit type of your DASD units and is defaulted to SYSDA.

Press the Enter key after you have finished keying in the order details. The Generate Jobstream panel appears.

Generate Jobstream Panel



```
CustomPac ----- Order Receive -----  
COMMAND ==>  
  
GENERATE JOBSTREAM  
  
Enter JOBCARDS  
  
> //FHOFMA JOB (XXXX),'PROGRAMMER NAME',  
> // CLASS=A,MSGCLASS=X,NOTIFY=FHOFMA  
> /*  
  
Installation ISPLLIB ==> ISP.V4R2M0.SISPEXEC  
           ==>  
ISPF ISPLLIB ==> ISP.V4R2M0.SISPMENU  
           ==>  
Libraries ISPLLIB ==> ISP.V4R2M0.SISPPENU  
           ==>  
           ISPSLIB ==> ISP.V4R2M0.SISPSENU  
           ==>  
           ISPTLIB ==> ISP.V4R2M0.SISPTENU  
           ==>
```

Figure 174. Generate Jobstream panel

5.2.5 Generate Jobstream panel

In this panel, enter the job card information relating to your installation standards. Change the ISPF library names to your current ISPF environment.

After pressing the Enter key, you enter a panel where you can specify additional job card information for loading of the RIMs. Depending on whether you have previously indicated that you wanted to edit the job stream before submission, you can now review and edit the generated job that is to receive the order, then submit it.

After successful completion of the job, your order data sets are copied from the RIM tape to your DASD.

After you have selected an order for processing, an enqueue is issued against the order number. This ensures that only one person can work on an individual order at any one time.

Installation Dialog Panel



```
CustomPac ----- (C) IBM Corporation 1990-1996 -----
OPTION ==> C

Installation

Order ( OS261466 ) The Following Functions MUST be Executed in Sequence

C  Configure      Select Configuration for Installation and Merge
*  Variables      Define Installation Variables
*  Zones          Define Zone Configuration
*  Modify         Modify System Layout
*  Alias          Define Alias to Catalog Relationships
*  SSA           Define SSA to Catalog Relationships
*  Installation   Select and Submit Installation Jobs
*  Save           Save Used Configuration
*  Update        Update Order Inventory Status
DI Display        Display a list of data set names
```

Figure 176. Installation Dialog panel

5.2.7 Installation dialog

After selecting a ServerPac order, the main installation dialog panel is invoked.

When this panel is shown for the first time, the only option which may be selected is option C. Each of the following functions now marked with an asterisk (*) become available after the previous function has successfully finished.

C Option C on this panel allows you to select a configuration for merging an initial installation. If this is your first CustomPac installation, the Create Configuration panel appears.

DI Option DI allows you to display any data set names and is similar to the PDF Option 3.4 function.

Create Configuration Panel



```
CustomPac ----- CREATE Configuration ( OS261466 ) ----- Row 1 to 1 of 1
COMMAND ==> CR                                     SCROLL ==> PAGE

Select Configuration

PRIM Cmds:(? SET L F N P SORT CReate)
LINE Cmds:(Select)

S CONFiguration          Comment
-----
*  SYSPOK.OS261466      Always Selected for Order
-----
*  SYSPOK.D96219       MASTER Configuration
-----
** NO MERGABLE ORDER CONFIGURATIONS
***** Bottom of data *****
```

Figure 177. Create Configuration panel

5.2.8 Selecting a configuration for the order

Before you start the installation, you must select and create a configuration. On the Create Configuration panel, you can see the master configuration and, if available, other saved configurations. If there is no previous configuration, you cannot merge with the current order.

Enter a *CR* in the command line to create a work configuration.

Type an *S* in front of the configurations you want to merge, if applicable.

Installation Variables Panel



```
CustomPac ----- Installation Variables ( OS261466 ) -- Row 1 to 14
COMMAND ==>                                SCROLL ==> PAGE

Variable Selection List                      SHOW( *      )

PRIM Cnds:(? SET L F N P CANCEL SAVE SHOW VARname)
LINE Cnds:(Browse Delete Edit Insert Repeat Ship)

S      Synonym          STA Contents
-----
==> GEN SYSTEMPAC
      ASSEMBLER NAME    D ASMA90
      ASSEMBLER NAME    D ASMA90
      ....
==> MVS OLD VOL
      VOLSER OLD RES     P OS3R2T
      VOLSER OLD CAT     D TOTCAT
      ....
```

Figure 178. Installation Variables panel

5.2.9 Define the installation variables

The Installation Variables panel can now be selected by typing a V option on the Installation panel. This takes you to the Installation Variables panel.

Verify the current contents and enter or change any values by overtyping in the Contents column if a value is either missing or invalid.

The installation variables can be different for each order. They are stored in the installation variables table (IVT), which is a CustomPac-generated ISPF table shipped with your order. The installation variables are briefly described in Appendix B (Variables used by this ServerPac) of the *ServerPac: Installing Your Order* publication that comes with your ServerPac.

It is recommended that you read and use the worksheets before changing any installation variable values.

The variable for AUTH.LINKLIB may be an existing authorized library of your installation site.

You may use the VARedit command on some panels to change the installation variables later.

Define ZONE Information Panel



CustomPac ----- Define ZONE Information (OS261466) --- Row 1 to 1 of 1
COMMAND ==> SCROLL ==> PAGE

PRIM Cmds:(? SET L F N P SORT CANcel SAVE SHIP)
LINE Cmds:(NONE)

ALL DLIB Zones will be allocated in CSI : SMPE.OS261466.CSI
ALL TLIB Zones will be allocated in CSI : SMPE.OS261466.CSI

For (Sub)System : MVS (Z038)

| NickName | Dlib Zone | Target Zone |
|----------|-----------|-------------|
| ----- | ----- | ----- |
| 100 | MVSD100 | MVST100 |
| 200 | MVSD200 | MVST200 |

Figure 179. Define ZONE Information panel

5.2.10 Defining SMP/E ZONE names

On the Installation Dialog panel, you may now select option Z to define your SMP/E zone configuration which brings you to the Define Zone Information panel.

This panel is displayed even if you do not plan to change the shipped zone names. You can change the zone names to the names you want. The nickname is used to pair them together.

The reason for having more than one target and DLIB zone is that you cannot have incompatible products together in one SMP/E zone, such as COBOL/II and OS/COBOL.

Use the SHIP command with caution, because it restores all DLIB and target zone names to their shipped value.

For more information, refer to *ServerPac: Installing Your Order*.

Modify System Layout Panel



```
CustomPac ----- Modify System Layout ( OS261466 ) -- Row 1 to 14
COMMAND ==>                                SCROLL ==> PAGE

SUMMARY Of Products

PRIM Cmds:(? SET F L N P SORT ALL CANcel SAVE SHIP DEVT SUML SUMP)
LINE Cmds:<Dslist Select>

  S Product                                VSLname  data sets
  - -----
  AD/CYCLE C/370                1.02.0   ADCE120   21
  AD/CYCLE CODE/370 ENU 1.02.0   ACCE120  13
  ADSM FOR MVS                   2.01.0   DSMB210  10
  ...
  OS/390 BCP                     O32B120  94
  D OS/390 ISPF                  O32F120  33
  OS/390 JES2                    OS3J120  15
  ...
```

Figure 180. Modify System Layout panel

5.2.11 Define system layout

Defining the target system layout is one of the most important steps during the order installation. You should use the Modify System Layout Sample Worksheet from Appendix A of *IBM ServerPac Using the Installation Dialog*, SC29-1244, before you enter any information on the Modify System Layout panel.

This panel shows you the summary of all products within your order. From here you start your customization of the individual products. You may modify the following information about the target, SMP/E, and catalog data sets:

- Product data set names, placement and attributes
- Logical volume to physical volume relationship
- Physical volumes device type, address, and volser

The *PRIM Cmds* and *LINE Cmds* on this panel give you greater flexibility in defining your new environment.

Read and use the section “Modify the System Layout” in Chapter 8 of *IBM ServerPac for OS/390 Using the Installation Dialog*, SC28-1244.

The *ServerPac: Installing Your Order* publication that comes with your order also contains all information relating to the products to be installed.

Data Set List by Product Panel



```
CustomPac ----- Modify System Layout ( OS261466 ) -- Row 1 to 14
COMMAND ==>                                SCROLL ==> PAGE

data set LIST PRODUCT : OS/390 ISPF

PRIM Ccmds:(? SET L F N P SORT CHange OFile OList VErbse)
LINE Ccmds:(Attribs Space)

S DSName                                F RECFM DSORG LRECL BLK
-----
ISP.AISPALIB                            FB    PO          80 88
ISP.AISPCLIB                            FB    PO          80 88
ISP.AISPEXEC                            FB    PO          80 88
ISP.AISPGENU                            FB    PO          80 88
ISP.AISPGMLI                            FB    PO          80 88
ISP.AISPGUI                             VB    PO         255 88
ISP.AISPHELP                            FB    PO          80 88
...
```

Figure 181. Data Set List by Product panel

5.2.12 Dslist line command

The Dslist line command displays the data list by product.

If you selected product OS/390 ISPF on the Modify System Layout main panel, all data sets for OS/390 ISPF are displayed. The PRIM command CHange allows you to make global changes to data set profiles. For example, you may change the HLQ for those product data sets. Before using the Change command, refer to the *IBM ServerPac for OS/390 Using the Installation Dialog*, SC28-1244, and read Chapter 8.

The line commands A and S allow you to change data set names, logical volumes, space, and BLKSIZE definitions for a specific data set profile.

Logical Volume by Product Panel



CustomPac ----- Modify System Layout (OS261466) ---- Row 1 to 1 of 1
COMMAND ==> SCROLL ==> PAGE

Logical Volume By PRODUCT

PRIM Cmds:(? SET L F N P SORT)

LINE Cmds:<Assign Dslist>

PRODUCT : OS/390 ISPF

| S | LVol | CYLs | Largest | data sets | PVol | DEV n |
|-----|--------|-------|---------|-----------|--------|-------|
| --- | ----- | ----- | ----- | ----- | ----- | ----- |
| | DLB067 | 128 | 35 | 16 | OS3DL1 | 3390- |
| | RES067 | 126 | 26 | 17 | OS3RS1 | 3390- |

***** Bottom of data *****

Figure 182. Logical Volume by Product panel

5.2.13 Select command

The Select command entered next to a product on the Modify System Layout main panel displays a summary of all logical volumes for the selected product.

The line command Assign allows you to assign all data set profiles for the selected logical volume to a different logical volume. LVol name DLBxxx stands for a DLIB Volume. LVol name RESxxx stands for a residence volume.

The line command Dslist displays all data sets for the selected logical volume. You can make global changes to the data set profiles as described with the Data Set List By Product panel.

List All User Defined Data Sets Panel



```
CustomPac ----- Modify System Layout ( OS261466 ) ---- Row 1 to 1 of 1
COMMAND ==>                                     SCROLL ==> PAGE

data set LIST All User Defined data sets

PRIM Cmds:(? SET L F N P SORT CHange OFile OList VErbose)
LINE Cmds:(Attribs Space Insert Delete)

S DSName                                     F RECFM DSORG LRECL BLK
-----
I
***** Bottom of data *****
```

Figure 183. List All User Defined Data Sets panel

5.2.14 Adding user-defined data sets

It is also possible to add your own user-defined data set profiles. To do this, return to the Modify System Layout main panel, then enter the PRIM Cmds ALL U.

The line command I displays a panel where you can define all the information needed to allocate a data set. See the *IBM ServerPac for OS/390 Using Installation Dialog*, SC28-1244, in Chapter 8 “Inserting a User Defined Data Set” for details.

Summary of Physical Volumes Panel



```

CustomPac ----- Modify System Layout ( OS261466 ) -- Row 1 to 1 of 1
COMMAND ==>                                     SCROLL ==> PAGE

SUMMARY Of Physical Volumes

PRIM Cmds:(? SET L F N P SORT DEVT)
LINE Cmds:<Assign Dslist>

  S PVol   CCUU  DEV name      CYLs  Largest  data set
  - - - - -
  OS3CAT  0CF1  3390-3      2198   261     21
  OS3DL1  0CF2  3390-3      2780   138     60
  OS3RS1  0CF0  3390-3      3010   170     63
  OS3RS2  OFF0  3390-3       942   254     20
***** Bottom of data *****
  
```

Figure 184. Summary of Physical Volumes panel

5.2.15 Display physical volumes

Before you leave the Modify System Layout main panel, you should enter the SUMP PRIM command, which displays a summary of the physical volumes.

When one of your physical volumes becomes over-allocated, the following message appears on the panel:

```
| CPP0605005S At least ONE PHYSICAL Volume is OVER ALLOCATED |
```

This condition is also shown by the <<<<<< next to the physical volume names.

By using the dialogs previously described, you are able to modify the system layout and correct the over-allocation of physical volumes.

Important

Use the SHIP command with care because it is powerful. This command is available on several dialog panels. It can be used to restore all profiles to their initial-ship values. You can lose all the customization you previously entered if you issue the SHIP command.

Define Catalog Data Set Name Panel



CustomPac ----- ALIAS to CATALOG (OS261466) ----- Row 1 to 14
 COMMAND ==> SCROLL ==> PAGE

Define CATALOG data set Names

PRIM Ccmds:(? SET L F N P SORT CANcel SAVE)
 LINE Ccmds:(Delete Insert Repeat)

| S | Alias | STA | TARGET System Catalog DSName | Type |
|---|--------|-----|------------------------------|------|
| | ADSM | | ???????.CATALOG | |
| | AOCIMS | | CATALOG.TOTICF1.VOS3RS2 | |
| | AOCOPE | | CATALOG.TOTICF1.VOS3RS2 | |
| | ASMA | M | CATALOG.TOTICFM.VOS3CAT | MCAT |
| | ASMT | | CATALOG.TOTICFM.VOS3CAT | MCAT |
| | BFS | M | CATALOG.TOTICFM.VOS3CAT | MCAT |
| | ... | | | |

Figure 185. Define Catalog Data Set Name panel

5.2.16 Define alias-to-catalog relationships

Specify the catalog data set name for each *alias*.

Before you begin to define the alias-to-catalog relationships and system-specific alias-to-catalog relationships, you should read Chapters 9 and 10 in *IBM ServerPac for OS/390 Using the Installation Dialog*, SC28-1244, to become familiar with using system specific-aliases (SSA) and the catalog structure.

Appendix A of *IBM ServerPac for OS/390 Using the Installation Dialog* contains the worksheets to be used for alias-to-catalog and SSA-to-catalog specifications.

Use the Alias to Catalog panel to specify which HLQ you want to be associated with a catalog. An *M* in the *STA* column indicates that this alias name must be associated with a master catalog.

The ?????? in the TARGET System Catalog DSName field indicates that there is no catalog defined yet.

This function allows you also to insert additional user-defined alias names and catalogs.

After specifying the alias-to-catalog relationship, you may select SSA on the Installation Dialog panel, which leads you to the SSA-to-Catalog panel.

SSA to Catalog Panel



```
CustomPac ----- SSA to CATALOG ( OS261466 ) ----- Row 1 to 1 of 1
COMMAND ==>                                     SCROLL ==> PAGE

CATALOG Selection List

PRIM Cnds:(? SET L F N P SORT CANCEL SAVE)
LINE Cnds:<Select>

S TARGET System Catalog DSName          SSA Name Type VOLser Alloc Def
-----
CATALOG.TOTICFM.VOS3CAT                 MOSR2   MCAT OS3CAT  Y  Y
CATALOG.TOTICF1.VOS3RS2                 UOSR2   UCAT OS3RS2  Y  Y
```

Figure 186. SSA to Catalog panel

5.2.17 Define system-specific alias names

The ServerPac installation process uses the system-specific alias (SSA) technique for data set allocation during the installation jobs. This allows you to work conveniently with new data sets that have same name as those on your existing system, for example SYS1.LINKLIB. We recommend you specify Y in the Def column for defining the SSAs. The SSAs are removed by a cleanup job after you have successfully installed the ServerPac.

This is the end of the customization steps for the ServerPac. You are now ready to run the supplied installation jobs.

Installation Jobs Panel



```
CustomPac ----- Installation JOBs ( OS261466 ) --- Row 1 to 14
COMMAND ==> GEN                                SCROLL ==> PAGE

JOB Selection List                               SS$( EXCL

PRIM Cmds:(? SET L F N P GENskel SUMmary SS$ VARedit)
LINE Cmds:(Backup Delete Edit Insert Log Output Select SS-block Vars)

S      Description                               STEP  MC STatus
-----
SRC DEFAULT JOBCARD
...
==> PACKAGE SPECIFIC INSTALL
DOC PACKAGE INSTALLATION STEPS
DOC OS/390 OE SYSTEM SERVICES
...
==> PRODUCT SPECIFIC INSTALL
DOC PRODUCT SPECIFIC INSTALL STEPS
DOC RMF 1.2.0
JOB ASSEMBLINK RMF FOR JES2                      HRM6602G 08 JOB14172
...
```

Figure 187. Installation Jobs panel

5.2.18 Run the ServerPac-provided installation jobs

There are three types of components shown on the Installation Jobs panel:

SRC Source data such as parameter lists

DOC Documentation

JOB Executable JCL

The installation steps are grouped into the following sections:

- Package-specific installation
- Product-specific installation
- Post-installation
- Additional post-installation
- Customization section
- Installation verification section
- Cleanup jobs
- Migration section
- Customer-specific customization

Generate File Tailored Jobs Panel



```
CustomPac ----- Installation JOBS ( OS261466 ) -----  
COMMAND ==>  
  
GENERATE File Tailored Installation Jobs  
  
This function generates a BATCH job which will file tailor  
ALL Installation Jobs in one pass, and save the jobs to the  
BACKUP data set.  
  
If a job already exists in the backup data set  
  
REPLACE Job ==> Y (Y or N)
```

Figure 188. Generate File Tailored Jobs panel

5.2.18.1 Generate File Tailored Jobs panel

When you enter the Installation Jobs panel for the first time, the installation jobs have still not been generated. All installation jobs are generated using ISPF tailoring services. The GENske1 command submits a batch job, which generates all the installation jobs. Each job is stored in the SCPPBENU data set that is provided through the SeverPac RECEIVE process.

The installation jobs should be submitted in sequence. Always read the DOC section before you select and submit the related jobs.

All installation steps and jobs are also described in *ServerPac: Installing Your Order*.

After a job's completion, the job output can be seen using the Output line command. This also updates the STATus column on the Installation Jobs panel.

The job, copying data sets to SystemPac Vols (RESTORE), may run for a long time, depending on the number of products your SeverPac order contains. You should have two tape drives and all the tape cartridges shipped with your order available before you start the RESTORE job.

Post-installation and customization is product- and installation-dependent, and should be related to your specific requirements.

You can insert your own defined jobs to these dialogs.

After the installation jobs have completed, you should be able to IPL and test your new OS/390 system.

Save Configuration Panel



CustomPac ----- SAVE Configuration (OS261466) -----
COMMAND ==>

Specify SAVE Library

Enter the High Level Qualifier(s) of the Library to which the
Order Configuration will be Saved

==> SYSPOK.OS261466.CONFIG

If the HLQ is NOT for the MASTER Configuration Library then
you may enter a Comment to Identify the Configuration

==> FIRST SETUP

MASTER HLQ is : SYSPOK.D96219

Figure 189. Save Configuration panel

5.2.19 Save Configuration panel

After the successful installation of your ServerPac, you may save your configuration by typing *S* on the main Installation Dialog Panel.

As the last step of the installation, you should update the inventory by entering a *U* on the main Installation Panel.

Appendix A. Fundamentals of OS/390

This appendix includes references from the Chapter on the Introduction to OS/390 Fundamentals.

A.1 List of Base Elements

Table 1 lists all elements that are in the base OS/390 system. The table tells you:

- Name** What this book calls the element.
- Exclusive (Ex)** Whether the element is exclusive. In the column, Yes identifies the exclusive elements (that is, having function available only in OS/390), and No identifies the nonexclusive elements (that is, the OS/390 level of an element is also available as a stand-alone product).
- Function Level (FL)** Specifies the function level in OS/390 (and in the stand-alone product); the latest OS/390 level in which the element changed ("Change" means that the element was added to OS/390, or one or more of its FMIDs was changed; new function added in PTFs is not considered change); and for nonexclusive elements, the equivalent level of the stand-alone product is listed in parentheses.

Note: Do not confuse the function level with the product level. All elements are at the R7 product level but they are at various function levels. For example, the product level of BDT is OS/390 R7 BDT. Its function level is OS/390 R2 BDT because R2 was the last release in which it changed.
- Comments** Miscellaneous facts that describe the element.

To learn what function the element provides, see *OS/390 Introduction and Release Guide*, GC28-1725.

| Table 1 (Page 1 of 9). Base Elements in OS/390 R7 | | | |
|---|-----|----|---|
| Name | Ex | FL | Comments |
| AET | Yes | R7 | <p>Application Enabling Technology (AET) is the base technology to build a ready-to-run, automated OS/390 UNIX System Services application server system, with all system programming already done. The built AET system does not require OS/390 skill to administer and operate it.</p> <p>AET was new to OS/390 in R3. It is related to the Automated UNIX System Option for VM, VSE, and OS/390 (Auto UNIX System) delivery option, 5655-A97. Auto UNIX System is a prebuilt, customized, and automated OS/390 UNIX System Services application server. It requires no OS/390 skills to install, use, maintain, and service. Traditional OS/390 applications such as TSO, CICS, IMS, and batch processing are not supported by Auto UNIX System.</p> |
| BCP | Yes | R7 | <p>The Base Control Program (BCP) provides essential operating system services. The BCP includes the I/O configuration program (IOCP) as well as the OS/390 UNIX System Services (OS/390 UNIX) kernel. This latter function was called OpenEdition System Services and was a nonexclusive base element of OS/390 R1, was an exclusive element of OS/390 R2, was integrated into the BCP element in OS/390 R3, and had its name changed to OS/390 UNIX System Services kernel in OS/390 R6.</p> |

Table 1 (Page 2 of 9). Base Elements in OS/390 R7

| Name | Ex | FL | Comments |
|------------------------------|-----|----|---|
| BDT | Yes | R2 | <p>Bulk Data Transfer (BDT) provides the base services that the optional BDT features (BDT File-to-File and BDT SNA NJE) need to transfer data from one computer system to another. BDT became exclusive in OS/390 R2.</p> <p>You cannot activate any BDT functions until one or both of the optional BDT features is enabled.</p> |
| BookManager BookServer | No | R4 | <p>BookManager BookServer converts BookServer BookManager books to HTML for display through a Web browser.</p> <p>BookManager BookServer was new to OS/390 in R4. Since then, the base hasn't changed but Japanese support was added in OS/390 R5 and Simplified Chinese support was added in OS/390 R7.</p> |
| BookManager READ | No | R1 | <p>BookManager READ is used to display, search, and manage online books and bookshelves. A related optional feature is BookManager BUILD.</p> |
| C/C++ IBM Open Class Library | Yes | R6 | <p>C/C++ IBM Open Class Library provides a set of C/C++ class libraries.</p> <p>As of OS/390 R6, the C/C++ IBM Open Class Library is a base element of OS/390. Retroactive to OS/390 R3, the C/C++ IBM Open Class Library is licensed with the OS/390 base operating system but is not considered a base element.</p> <p>The above changes enable your applications to use the C/C++ IBM Open Class Library at run time without having to license either the C/C++ with Debug Tool or C/C++ without Debug Tool features. These changes also enable your applications to access the required dynamic link libraries (DLLs) so you do not have to use the DLL Rename Utility to package and redistribute these DLLs with your applications.</p> |
| Cryptographic Services | Yes | R7 | <p>Cryptography is the transformation of data to conceal its meaning. In OS/390, the base element Cryptographic Services provides the following base cryptographic functions: data secrecy, data integrity, personal identification, digital signatures, and the management of cryptographic keys. Additional cryptographic functions are provided by the following related optional features:</p> <ul style="list-style-type: none"> * Open Cryptographic Services Facility (OCSF) France * OCSF Security Level 1 * OCSF Security Level 2 * OCSF Security Level 3 * System Secure Sockets Layer (SSL) Crypto <p>Cryptographic Services was new in OS/390 R7. It includes the Integrated Cryptographic Service Facility (ICSF), which was a new base element in OS/390 R4 and became exclusive in OS/390 R6.</p> |
| DCE Application Support | Yes | R7 | <p>DCE Application Support facilitates the interaction between Distributed Computing Environment (DCE) clients and CICS or IMS regions. This element was new to OS/390 in R4. It is based on stand-alone product OpenEdition DCE Application Support for MVS/ESA R2.</p> <p>As of OS/390 R6, the word "OpenEdition" was dropped from the beginning of this element's name.</p> |

| <i>Table 1 (Page 3 of 9). Base Elements in OS/390 R7</i> | | | |
|--|-----------|-----------|---|
| Name | Ex | FL | Comments |
| DCE Base Services | Yes | R7 | DCE Base Services provides services for developing and running client/server applications, including remote procedure call, directory, security, and distributed time services. This element is at the Open Group Open Software Foundation (OSF) DCE 1.1 level. As of OS/390 R6, the word "OpenEdition" was dropped from the beginning of this element's name. |
| DFSMSdfp | No | R7 | DFSMSdfp provides storage, data, program, and device management functions. Related optional features are: DFSMSrmm, DFSMSdss, and DFSMShsm. |
| Distributed File Service | Yes | R7 | Distributed File Service includes the file serving component of the OSF DCE. The DCE file serving support is at the OSF 1.2.2 level. Prior to OS/390 R5, this element was called OpenEdition DCE DFS. |
| Encina Toolkit Executive | Yes | R7 | Encina Toolkit Executive provides a set of tools for developing client components of distributed transactional applications. This element was new to OS/390 in R4. |

Table 1 (Page 4 of 9). Base Elements in OS/390 R7

| Name | Ex | FL | Comments |
|--------------------------------|-----|----|---|
| eNetwork Communications Server | Yes | R7 | <p>eNetwork Communications Server (also known as CS for OS/390 and SecureWay Communications Server) supports secure TCP/IP, SNA, and UNIX networking throughout an enterprise. It gives you the ability to connect subsystems and applications to each other, and to connect network devices (such as terminals and printers) to the system.</p> <p>eNetwork Communications Server consists of two components: IP and SNA. SNA includes AnyNet function. Prior to OS/390 R6, IP was the base element TCP/IP and SNA was the base element VTAM.</p> <p>Related optional features are:</p> <ul style="list-style-type: none"> * eNetwork Communications Server Security Level 1 * eNetwork Communications Server Security Level 2 * eNetwork Communications Server Security Level 3 * eNetwork Communications Server NPF <p>IP can be dynamically enabled. If you order the standard OS/390 base, IP is shipped enabled. But if you order the alternate base (see "OS/390 Alternate Base" in topic 4.1.1.1), IP is shipped disabled.</p> <p>Many of the IP functions have changed over the life of OS/390, as follows:</p> <ul style="list-style-type: none"> * TCP/IP OpenEdition. In OS/390 R1, R2, and R3, this was an optional feature called TCP/IP for OpenEdition MVS Applications. In OS/390 R4 it became part of the TCP/IP element and a new stack was added. * TCP/IP CICS Sockets. In OS/390 R1 and R2, this was an optional feature called TCP/IP for MVS CICS Sockets. In OS/390 R3 it became part of the TCP/IP element. * TCP/IP IMS Sockets. In OS/390 R1 and R2, this was an optional feature called TCP/IP for MVS IMS Sockets. In OS/390 R3 it became part of the TCP/IP element. * Host On-Demand. This was new to OS/390 in R4. Note that there is a stand-alone product, IBM eNetwork Host On-Demand V3 (part number 31L2157, 31L2158, or 31L2159; order type number 5648-B40), which contains additional function. * X-Windows. In OS/390 R5, the older X-Windows function (X11R4) was taken out of the TCP/IP base and put into a separate FMID. New X-Windows function (X11R6) was added to the base. * DNS/WLM. This function, the Domain Name Server with Workload Manager, was available as a kit for OS/390 R4 and then was integrated into OS/390 R5. * Network Station Client and Network Station Manager. As of OS/390 R5, these functions are no longer in OS/390. They are available as the stand-alone product Network Station Manager (5648-C05). * 3172 Offload. As of OS/390 R5, this function is no longer in OS/390. |

| <i>Table 1 (Page 5 of 9). Base Elements in OS/390 R7</i> | | | |
|--|-----------|-----------|---|
| Name | Ex | FL | Comments |
| EREP | No | R1 | The Environmental Record Editing and Printing Program (EREP) edits and prints reports for the records placed in the error recording data set (ERDS), helping IBM service representatives fix problems. |
| ESCON Director Support | No | R1 | ESCON Director Support enables the reporting of ESCON director device errors to OS/390. This element was an orderable feature of MVS/ESA SP 4.2 and later releases. |
| FFST | No | R2 | First Failure Support Technology (FFST) provides immediate notification and first failure data capture for software events. FFST was new to OS/390 in R2. |
| GDDM | No | R2 | GDDM provides presentation services and device-driving capability. It includes PCLK and OS/2 Link and REXX code. Related optional features are GDDM-Presentation Graphics Feature and GDDM-REXX. Other GDDM-associated products (IVU, GKS, IMD) are not in OS/390, but are separately orderable. |
| HCD | Yes | R5 | Hardware Configuration Definition (HCD) defines both the operating system configuration and the processor hardware configuration for a system. A related optional feature is HCM. |
| HLASM | No | R7 | High Level Assembler (HLASM) integrates almost all functions of past assemblers and provides extensions and improvements. A related optional feature is the HLASM Toolkit. |
| ICKDSF | No | R1 | The Device Support Facility (ICKDSF) enables you to perform functions needed for the installation and use of IBM DASD. |
| ISPF | Yes | R5 | ISPF is a full-screen editor and dialog manager. As of OS/390 R5, ISPF is exclusive to OS/390. |
| JES2 | Yes | R7 | JES2 accepts the submission of work for the BCP. This element exercises independent control over its job processing functions, whereas JES3 exercises centralized control. JES2 4.2, JES2 4.3, JES2 5.1, JES2 5.2, and OS/390 levels of JES2 are supported on a single OS/390 R7 system. |
| LAN Server | Yes | R5 | LAN Server enables LAN workstation users to store and share data and applications in a central location on a System/390, which allows the large storage capacity of a System/390 to relieve the capacity constraints of workstation-based servers. This element can be used as a file-sharing system for OS/2 LAN Server networks, a Network File System file-serving protocol, or both. Support for Network File System file-serving protocols is provided either through TCP/IP or through Network File System front-end processors. LAN Server and Network File System Server cannot work simultaneously on the same processor to provide Network File System services through TCP/IP. At setup or run time, you must choose which function to use. The host portion of LAN Server is shipped on tape. The workstation portion is shipped on diskettes. Prior to becoming an exclusive element of OS/390, LAN Server was part of MVS/ESA SP V5R2. It was also called LAN File Services (LFS). As of OS/390 R3, LAN Server English and Kanji features can coexist on the same system and in the same target zone. |

Table 1 (Page 6 of 9). Base Elements in OS/390 R7

| Name | Ex | FL | Comments |
|----------------------|-----|----|--|
| Language Environment | Yes | R7 | <p>Language Environment provides the run-time environment for programs generated with:</p> <ul style="list-style-type: none"> * OS/390 C/C++ * C/C++ for MVS/ESA * AD/Cycle C/370 * VisualAge for Java, Enterprise Edition for OS/390 * AD/Cycle C/370 * COBOL for OS/390 and VM * COBOL for MVS and VM * AD/Cycle COBOL/370 * PL/I for MVS and VM * AD/Cycle PL/I for MVS and VM * VS FORTRAN and FORTRAN IV (in compatibility mode) <p>Prior to OS/390, this element was known as Language Environment for MVS and VM or AD/Cycle Language Environment/370 (LE/370).</p> <p>For more information on IBM VisualAge for Java, Enterprise Edition for OS/390, program number 5655-JAV, refer to the product documentation.</p> <p>Inclusion of Language Environment for MVS into OS/390 does not replace the need for separate compilers.</p> <p>A function of the BCP called run-time library services (RTLS) allows you to use run-time options to control access to different levels of the Language Environment run-time libraries. RTLS can be used to access the run-time library of Language Environment V1R5 and all OS/390 levels of Language Environment.</p> <p>A related optional feature is Language Environment Data Decryption.</p> |
| LANRES | Yes | R5 | <p>LANRES integrates NetWare LANs and System/390 environments. Included with LANRES are diskettes that you use to install LANRES code on the Novell NetWare Server.</p> |
| MICR/OCR | No | R1 | <p>This element provides the device support code for various magnetic and optical devices.</p> |
| NetQuestion | Yes | R6 | <p>NetQuestion is a text search engine for use with the element WebSphere Application Server. NetQuestion was new to OS/390 in R4.</p> <p>NetQuestion is described in the Webserver Search Engine publications.</p> |

| <i>Table 1 (Page 7 of 9). Base Elements in OS/390 R7</i> | | | |
|--|-----------|-----------|---|
| Name | Ex | FL | Comments |
| Network File System | Yes | R6 | <p>Network File System acts as a file server to workstations, personal computers, or other authorized systems in a TCP/IP network. In OS/390 R2 it was enhanced and its function was equivalent to the stand-alone product DFSMS/MVS V1R3 Network File System. In OS/390 R6 it became exclusive to OS/390, function was added, and "DFSMS/MVS" was dropped from the beginning of its name.</p> <p>This element consists of a client (Network File System Client) and a server (Network File System Server).</p> <p>Network File System Server and LAN Server cannot work simultaneously on the same processor to provide Network File System services through TCP/IP. At setup or run time, you must choose which function to use.</p> <p>This element supports Berkeley Sockets, and not TCP/IP Sockets.</p> <p>As of OS/390 R4, this element is always enabled, even when the OS/390 alternate base configuration is ordered.</p> |
| OSA/SF | No | R2 | <p>Open Systems Adapter Support Facility (OSA/SF) customizes the modes and port parameters of an OSA-2, which provides S/390 network connectivity directly to LANs and WANs that support IP and SNA protocols. The OSA-2 must be defined as an S/390 channel path with integrated network ports.</p> <p>As of OS/390 R6 (via PTF UW52168), multiple home IP addresses can be specified per data path (OSA table entry) through the OSA-2, as well as additional redundant (default LP) pathing. IP multicasting is supported. For a Fast Ethernet (FENET) OSA-2 only, IPX/SPX protocols are supported in the OSA-2 HPDT MPC mode.</p> |
| SMP/E | Yes | R7 | <p>SMP/E is a tool for installing and maintaining software, and for managing the inventory of software that has been installed.</p> <p>The Planning and Migration Assistant, a component of SMP/E, can help you maintain, plan for, and order new releases of OS/390 and other products. It provides reports that use IBM-supplied data, your SMP/E CSI data set, and a CustomPac inventory file. The Planning and Migration Assistant is part of OS/390 R3, R4, R5, and R6 via PTFs, and is integrated into OS/390 as of R7. It was enhanced in OS/390 R7. The Planning and Migration Assistant Web site is http://www.ibm.com/s390/pma/</p> |

Table 1 (Page 8 of 9). Base Elements in OS/390 R7

| Name | Ex | FL | Comments |
|-----------------------------|-----|----|---|
| Softcopy Print | Yes | R7 | <p>Softcopy Print allows you to print BookManager books. This element has a base, single-byte character set (SBCS) function plus a double-byte character set (DBCS) function. The DBCS function ships with DBCS national language versions of OS/390 (Simplified Chinese, Traditional Chinese, Japanese/Kanji, and Korean).</p> <p>This element consists of integrated subsets of PSF V3R1 for OS/390 (called PSF for Softcopy Print), Document Composition Facility (DCF) V1R4, BookMaster V1R4, and some fonts in the AFP Font Collection V1R1. Softcopy Print for DBCS adds the DBCS Print Utility, DCF DBCS, and BookMaster Gothic fonts.</p> <p>The base SBCS function was new in OS/390 R2. The DBCS function was new in OS/390 R4.</p> <p>In a related matter, as of OS/390 R7, the PSF stand-alone product that is supported on OS/390 is PSF V3 for OS/390 (5655-B17), not PSF/MVS V2 (5695-040). (PSF/MVS V2 is still supported on OS/390 R6 and down.) In addition, in order to use PSF V3 for OS/390, you must install the complete product. The product was repackaged to isolate the Softcopy Print functions and, unlike PSF/MVS V2, it is not possible to dynamically enable PSF V3 for OS/390.</p> |
| SOMobjects | Yes | R4 | <p>SOMobjects is a technology that RTL allows applications written in different programming languages to use the same object-oriented class libraries. The base element SOMobjects Runtime Library (RTL) is a set of functions for creating objects and invoking methods on them.</p> <p>Prior to becoming an exclusive element of OS/390, SOMobjects RTL was part of MVS/ESA SP 5.2.2 and a feature of the product SOMobjects for MVS V1R1.</p> <p>A related optional feature is SOMobjects ADE.</p> |
| TIOC | No | R1 | TIOC allows console services and TSO/E to communicate with the terminal hardware. |
| Tivoli Management Framework | No | R7 | Tivoli Management Framework enables OS/390 to be managed (via an agent) by the Tivoli Framework-based applications that support OS/390. This element was new to OS/390 in R7. |
| TSO/E | Yes | R4 | Time Sharing Option/Extended (TSO/E) provides an interactive terminal interface. As in prior releases of TSO/E, this element includes CLISTS and REXX, but does not include a REXX compiler. |
| OS/390 UNIX System Services | Yes | R7 | OS/390 UNIX System Services Application Services provides the services and a standard command interface familiar to interactive UNIX users. In OS/390 R2, the OpenEdition MVS Debugger and OpenEdition MVS Shell and Utilities were merged into this element. In OS/390 R6, this element's name changed from OpenEdition Application Services to OS/390 UNIX System Services Application Services. |
| VisualLift RTE | Yes | R1 | <p>VisualLift Runtime Environment (RTE) and the optional feature VisualLift Application Development Environment (ADE) are tools to modernize the user interface of existing host applications.</p> <p>Prior to OS/390, both the RTE and the ADE were packaged together as the VisualLift product. With OS/390, VisualLift is repackaged; VisualLift RTE is a base element and VisualLift ADE is an optional feature.</p> <p>When you install VisualLift RTE you will first install the code on the host, and then download code from the host to the workstation and install VisualLift RTE there.</p> |

| <i>Table 1 (Page 9 of 9). Base Elements in OS/390 R7</i> | | | |
|--|-----------|-----------|---|
| Name | Ex | FL | Comments |
| WebSphere Application Server | Yes | R7 | <p>WebSphere Application Server is a scalable, high-performance Web server. As of OS/390 R7, it is exclusive to OS/390.</p> <p>In OS/390 R5 and R6, this element was named Lotus Domino Go Webserver. Prior to OS/390 R5, its name was Internet Connection Secure Server (ICSS).</p> <p>WebSphere Application Server includes the component ServletExpress. ServletExpress is a portable, Java servlet-based execution environment that extends WebSphere Application Server into a Java Web Application Server. ServletExpress was new to OS/390 as of OS/390 R6. (Outside of OS/390, ServletExpress was new in the Domino Go Webserver 4.6.1 stand-alone product.)</p> <p>In order to have secure communication, one of the following optional features must be installed:</p> <ul style="list-style-type: none"> * IBM HTTP Server Export Secure * IBM HTTP Server France Secure * IBM HTTP Server NA Secure <p>WebSphere Application Server uses the base element NetQuestion as its text search engine.</p> |
| 3270 PC File Transfer Program | No | R2 | 3270 PC File Transfer Program transfers files from the host to the workstation for off-line data manipulation, updating, or correction or for the transfer and storage of local data in the host stem. This element was new to OS/390 in R2. |

A.2 List of Optional Elements

Table 2 summarizes OS/390 optional features. As in Table 1 on page 263, Ex identifies the exclusive features and FL identifies the latest OS/390 level in which the feature changed.

| <i>Table 2 (Page 1 of 6). Optional Elements in OS/390 R7</i> | | | |
|--|-----------|-----------|--|
| Name | Ex | FL | Comments |
| BDT File-to-File | Yes | R2 | This feature allows users at one OS/390 system in a SNA network to copy data sets to or from another OS/390 system in the network. This feature is related to the element BDT. |
| BDT SNA NJE | Yes | R2 | This feature allows JES3 users to transmit jobs, output, commands, and messages from one computer system to another within a SNA network. This feature is related to the element BDT and the feature JES3. |
| BookManager BUILD READ | No | R1 | BookManager BUILD creates online books to be used by BookManager |

Table 2 (Page 2 of 6). Optional Elements in OS/390 R7

| Name | Ex | FL | Comments |
|--------------------------------|-----|----|---|
| C/C++ with Debug Tool | Yes | R6 | <p>This feature includes:</p> <ul style="list-style-type: none"> • A C compiler • A C++ compiler • A debug tool, which runs with C and C++ as well as other languages • C/C++ application development utilities <p>This feature is related to the base element C/C++ IBM Open Class Library.</p> <p>The C/C++ Database Access Class Library (DACL) Utility was removed in OS/390 R4.</p> |
| C/C++ Without Debug Tool | Yes | R6 | <p>This feature is the same as the C/C++ above except that it does not have the Debug Tool component.</p> |
| DCE User Data Privacy CDMF | Yes | R6 | <p>This feature enables data encryption using the commercial data masking facility (CDMF) algorithm. In OS/390 R6, the word "OpenEdition" was dropped from the beginning of this feature's name.</p> |
| DCE User Data Privacy DES/CDMF | Yes | R6 | <p>This feature enables data encryption using the data encryption standard (DES) algorithm and the (CDMF) DES/CDMF algorithm.</p> <p>The availability of this feature outside the United States is subject to United States export regulations.</p> <p>In OS/390 R6, the word "OpenEdition" was dropped from the beginning of this feature's name.</p> |
| DFSMSdss | No | R4 | <p>This feature is a DASD data and space management tool.</p> <p>Before becoming a component of stand-alone product DFSMS/MVS, DFSMSdss was a product called Data Facility Data Set Services (DFDSS) V2R5.</p> |
| DFSMShsm | No | R4 | <p>This feature is a DASD storage management and productivity tool for managing low-activity and inactive data.</p> <p>Before becoming a component of DFSMS/MVS, DFSMShsm was a product called Data Facility Hierarchical Storage Manager (DFHSM).</p> |
| DFSMSrmm | No | R4 | <p>This feature helps you manage your removable media as one enterprise-wide library across systems that can share DASD.</p> |
| DFSORT | No | R2 | <p>This feature sorts, merges, and copies data. This feature was new to OS/390 in R2.</p> <p>During OS/390 R3, DFSORT PTFs were released that changed its packaging. These PTFs were incorporated in OS/390 R4 DFSORT and can also be installed in the stand-alone DFSORT V1R13 product. Resident and nonresident installation is combined; you can now specify whether DFSORT modules are to be LPA-resident using the LPA list. Also, all DFSORT language features (the ISPF and ISMF U.S. English and Japanese (Kanji) panels and messages) can now be installed at the same time. You determine whether the panels are to be used and which language will be used by concatenating the proper libraries to existing ISPF libraries.</p> <p>The next release of the stand-alone product (5740-SM1), V1R14, may be ordered by OS/390 customers without an additional fee. This level will be integrated into OS/390 R7.</p> |

| <i>Table 2 (Page 3 of 6). Optional Elements in OS/390 R7</i> | | | |
|--|-----------|-----------|--|
| Name | Ex | FL | Comments |
| Domino Go Webserver Export Security | No | R6 | <p>This feature includes Secure Sockets Layer (SSL), Proxy Authentication, and a repository for home pages. It supports multiple IP addresses and double-byte character set. It is subject to United States export regulations and has import restrictions into France.</p> <p>Before OS/390 R5, this feature was called ICSS Export Security. As of OS/390 R5 it was rebranded to Lotus Domino Go Webserver Export Security.</p> <p>This feature is related to the base element Domino Go Webserver.</p> |
| Domino Go Webserver France Secure | No | R6 | <p>This feature includes Secure Sockets Layer (SSL), Proxy Authentication, and a repository for home pages. It supports multiple IP addresses and double-byte character set. It is subject to United States export regulations and may be exported into France.</p> <p>This feature was new in OS/390 R5.</p> <p>This feature is related to the base element Domino Go Webserver.</p> |
| Domino Go Webserver North America Secure | No | R6 | <p>This feature includes Secure Sockets Layer (SSL), Proxy Authentication, and a repository for home pages. It supports multiple IP addresses and double-byte character set. It is subject to United States export regulations and may not be exported from the United States or Canada.</p> <p>Before OS/390 R5, this feature was called ICSS North America Secure. As of OS/390 R5 it was rebranded to Lotus Domino Go Webserver North America Secure.</p> <p>This feature is related to the base element Domino Go Webserver.</p> |
| GDDM-PGF | No | R2 | <p>GDDM-Presentation Graphics Feature (PGF) is a set of programs for creating presentation material in a variety of styles. This feature was new to OS/390 in R2.</p> <p>This feature is related to the base element GDDM.</p> |
| GDDM-REXX | No | R2 | <p>This feature is a productivity tool that enables programmers to prototype GDDM applications and to create small routines and utility programs quickly and easily.</p> <p>This feature is related to the base element GDDM.</p> |
| HCM | Yes | R4 | <p>Hardware Configuration Manager (HCM) is a PWS-based client/server interface to the base element HCD. This feature was new to OS/390 in R4.</p> |
| HLASM Toolkit | No | R1 | <p>This feature provides tools to improve application development, debugging, and recovery. It is related to the base element HLASM.</p> <p>Customers licensed for the OS/390 HLASM Toolkit feature may order a copy of the next release of the feature prior to its availability by ordering the Toolkit feature of the V1R3 stand-alone product, just as they may order the stand-alone product.</p> |

Table 2 (Page 4 of 6). Optional Elements in OS/390 R7

| Name | Ex | FL | Comments |
|--------------------------------------|-----|----|---|
| IP Security - CDMF | Yes | R6 | <p>This feature provides support for packet filtering, tunnels, and network address translation (NAT), which enables secure communication over private and public networks.</p> <p>This feature includes Secure Sockets Layer (SSL) RC2/RC4.</p> <p>This feature started out as a function in the OS/390 R4 Firewall Technologies kit, was integrated into OS/390 R5 as a feature of OS/390, and was updated in OS/390 R6. It is related to the base element eNetwork Communications Server, called SecureWay Communications Server in Release 8.</p> |
| IP Security - DES/CDMF | Yes | R6 | <p>This feature provides support for packet filtering, tunnels, and network address translation (NAT), which enables secure communication over private and public networks. This feature includes data encryption standard (DES) encryption, which is stronger encryption than that found in IP Security - CDMF.</p> <p>This feature includes SSL DES.</p> <p>The availability of this feature outside the United States is subject to United States export regulations.</p> <p>This feature started out as a function in the OS/390 R4 Firewall Technologies kit, was integrated into OS/390 R5 as a feature of OS/390, and was updated in OS/390 R6. It is related to the base element eNetwork Communications Server, called SecureWay Communications Server in Release 8.</p> |
| IP Security - TDES | Yes | R6 | <p>This feature provides support for packet filtering, tunnels, and network address translation (NAT), which enables secure communication over private and public networks. This support uses DES encryption.</p> <p>This feature includes SSL triple DES (TDES) for secure communication between a TN3270 server and an SSL-enabled client. This is stronger encryption than that found in IP Security - DES/CDMF.</p> <p>The availability of this feature outside the United States is subject to United States export regulations.</p> <p>This feature was new to OS/390 R6. It is related to the base element eNetwork Communications Server, called SecureWay Communications Server in Release 8.</p> |
| JES3 | Yes | R6 | <p>JES3 accepts the submission of work for the BCP. JES3 exercises centralized control over its job processing functions, whereas JES2 exercises independent control.</p> <p>JES3 4.2.1, JES3 5.1.1, JES3 5.2.1, and OS/390 levels of JES3 were supported as of OS/390 R6.</p> |
| Language Environment Data Decryption | Yes | R6 | <p>This feature provides decryption of data using the DES algorithm for use with certain C functions. This feature is subject to United States export regulations.</p> |

Table 2 (Page 5 of 6). Optional Elements in OS/390 R7

| Name | Ex | FL | Comments |
|----------------------------|-----|----|--|
| OS/390 Print Server | Yes | R6 | <p>OS/390 Print Server allows you to print files on OS/390 printers from any workstation that has TCP/IP access. This feature was new to OS/390 in R5. It consists of three components:</p> <ul style="list-style-type: none"> * IP PrintWay. This component is also a feature of PSF/MVS V2R2 and first became part of OS/390 as part of the IP PrintWay/NetSpool feature of OS/390 R3. The base FMID is unchanged since OS/390 R3. Japanese support was added in OS/390 R5 and Spanish support was added in OS/390 R6. * NetSpool. This component is also a feature of PSF/MVS V2R2 and first became part of OS/390 as part of the IP PrintWay/NetSpool feature of OS/390 R3. The base FMID is unchanged since OS/390 R3. Japanese support was added in OS/390 R5 and Spanish support was added in OS/390 R6. * OS/390 Print Interface. This component was new in OS/390 R5 and is exclusive to OS/390. In OS/390 R6, the only change was that Japanese and Spanish support were added. <p>OS/390 Print Server replaces the IP PrintWay/NetSpool feature of OS/390. In addition, just as the IP PrintWay/NetSpool feature was recommended for use instead of the TCP/IP NPF feature in OS/390 R4, OS/390 Print Server in OS/390 R6 is recommended over TCP/IP NPF. The IP PrintWay component provides improved function, capacity, performance, and usability over NPF.</p> |
| RMF | Yes | R6 | Resource Measurement Facility (RMF) gathers data about OS/390 resource usage and provides reports about any system in a sysplex. |
| SDSF | YES | R5 | System Display and Search Facility (SDSF) provides you with information to monitor, manage, and control your OS/390 system. This feature was new to OS/390 in R2. |
| Security Server | Yes | R6 | <p>Security Server lets you control access to protected resources. Security Server consists of these components:</p> <ul style="list-style-type: none"> * RACF, which became exclusive in OS/390 R3. * DCE Security Server, which was exclusive as of OS/390 R1. This component was upgraded to support Kerberos V5.1 in OS/390 R5. * LDAP Server, which was new in OS/390 R5. This component provides a directory service based on the Lightweight Directory Access Protocol (LDAP), allowing clients to search, extract, add, and delete information from an LDAP server running on OS/390. * Firewall Technologies, which was part of the OS/390 R4 Firewall Technologies kit and integrated into OS/390 R5. Note that you need to order the IP Security - CDMF feature, the IP Security - DES/CDMF feature, or the IP Security - TDES feature to complete the firewall security package. |
| Security Server LDAP - DES | Yes | R6 | <p>This feature provides cryptographic protection above what is provided by the LDAP Server component of the Security Server feature. It contains DES/TDES function.</p> <p>This feature was new in OS/390 R5, is orderable only by customers who order the Security Server feature, is subject to United States export regulations, and may not be exported from the United States or Canada.</p> |

Table 2 (Page 6 of 6). Optional Elements in OS/390 R7

| Name | Ex | FL | Comments |
|-------------------------|-----|----|---|
| SOMobjects ADE | Yes | R4 | <p>SOMobjects Application Development Environment (ADE) provides a SOM compiler and source code for the SOM kernel (root) classes, Interface Repository Framework, and emitter Framework.</p> <p>This feature is related to the base element SOMobjects RTL.</p> |
| TCP/IP Kerberos DES | Yes | R6 | <p>This feature provides authentication and security services in a TCP/IP network environment. It uses the DES algorithm to encrypt data. This feature is exclusive to OS/390 as of OS/390 R5. The availability of this feature outside the United States is subject to United States export regulations.</p> |
| TCP/IP Kerberos non-DES | Yes | R6 | <p>This feature provides authentication and security services in a TCP/IP network environment. It uses a less stringent encryption algorithm than the DES algorithm. This feature is exclusive to OS/390 as of OS/390 R5.</p> |
| TCP/IP NPF | Yes | R6 | <p>This feature provides a printing function that reroutes print data to the IP network. This feature is exclusive to OS/390 as of OS/390 R5.</p> <p>OS/390 Print Server should be considered IBM's strategic replacement for TCP/IP Network Print Facility (NPF). Significant future enhancements are not planned for TCP/IP NPF.</p> |
| VisualLift ADE | No | R2 | <p>VisualLift Application Development Environment (ADE) and the base element VisualLift Runtime Environment (RTE) are tools to modernize the user interface of existing host applications.</p> <p>VisualLift ADE is the only priced feature that does not support dynamic enablement. It is shipped (on diskette) only if you order it.</p> |

Appendix B. System Programmer Customization Examples

This appendix includes references from the Chapter on the Introduction to OS/390 system programmer management of functions.

B.1 SYS1.PARMLIB Member IEFSSNxx

```
SUBSYS SUBNAME(SMS)          /* SMS */                00010000
  INITRTN(IGDSSIIN)          00010100
  INITPARM(' ID=00,PROMPT=NO') 00010200
SUBSYS SUBNAME(&PRISUBSY)    /* PRIMARY SUBSYSTEM NAME */ 00010300
  PRIMARY(YES) START(NO)     00010400
SUBSYS SUBNAME(&SECSUBSY)    /* SECONDARY SUBSYSTEM */    00010500
SUBSYS SUBNAME(RRS)         /* RESOURCE RECOVERY SERVICES */ 00010620
SUBSYS SUBNAME(DFRM)        /* DFSMSRMM */                00010720
  INITRTN(EDGSSSI)          00010820
SUBSYS SUBNAME(IRLM)         /* IMS RESOURCE LOCK MANAGER */ 00011706
SUBSYS SUBNAME(IRLS)         /* IMS RESOURCE LOCK MANAGER */ 00011826
SUBSYS SUBNAME(JRLM)         /* SECONDARY SUBSYSTEM NAME FOR IRLM */ 00011926
SUBSYS SUBNAME(IRL5)         /* IRLM DB2 5.1 */                00012026
SUBSYS SUBNAME(IRLT)         /* IRLM DB2 5.1 TEST */        00012128
SUBSYS SUBNAME(IRLA)         /* IRLM DB2 5.1 FOR DB2V510A */    00012234
SUBSYS SUBNAME(IRLB)         /* IRLM DB2 5.1 FOR SC48 */        00012334
SUBSYS SUBNAME(IRLC)         /* IRLM DB2 5.1 FOR DB2V510C */    00012439
SUBSYS SUBNAME(IRLD)         /* IRLM DB2 5.1 FOR DB2V510D */    00012539
SUBSYS SUBNAME(IRLE)         /* IRLM DB2 5.1 FOR DB2V510E */    00012642
SUBSYS SUBNAME(IRLF)         /* IRLM DB2 5.1 FOR DB2V510F */    00012745
SUBSYS SUBNAME(IRLG)         /* IRLM DB2 5.1 FOR DB2V510G */    00012845
SUBSYS SUBNAME(IRLH)         /* IRLM DB2 5.1 FOR DB2V510H */    00012947
SUBSYS SUBNAME(IRLI)         /* IRLM DB2 5.1 FOR DB2V510I */    00013048
SUBSYS SUBNAME(IRLJ)         /* IRLM DB2 6.1 FOR DB2V610J */    00013149
SUBSYS SUBNAME(IRLK)         /* IRLM DB2 5.1 FOR DB2V510K */    00013254
SUBSYS SUBNAME(IRLL)         /* IRLM DB2 5.1 FOR DB2V510L */    00013355
SUBSYS SUBNAME(IRLN)         /* IRLM DB2 5.1 FOR DB2V510N */    00013456
SUBSYS SUBNAME(IRLV)         /* IRLM DB2 5.1 FOR DB2V510V */    00013557
SUBSYS SUBNAME(IR6C)         /* IRLM IMS 6.1 FOR IMS610C */    00013657
SUBSYS SUBNAME(JR6C)         /* IRLM IMS 6.1 FOR IMS610C */    00013757
SUBSYS SUBNAME(IR6D)         /* IRLM IMS 6.1 FOR IMS610D */    00013857
SUBSYS SUBNAME(JR6D)         /* IRLM IMS 6.1 FOR IMS610D */    00013957
SUBSYS SUBNAME(IR6P)         /* IRLM IMS 6.1 FOR IMS6PEG */    00014057
SUBSYS SUBNAME(JR6P)         /* IRLM IMS 6.1 FOR IMS6PEG */    00014157
SUBSYS SUBNAME(IR6S)         /* IRLM IMS 6.1 FOR IMS610S */    00014257
SUBSYS SUBNAME(JR6S)         /* IRLM IMS 6.1 FOR IMS610S */    00014357
SUBSYS SUBNAME(IR6T)         /* IRLM IMS 6.1 FOR IMS610T */    00014457
SUBSYS SUBNAME(JR6T)         /* IRLM IMS 6.1 FOR IMS610T */    00014557
SUBSYS SUBNAME(DB41)        00014657
  INITRTN(DSN3INI)          00014757
  INITPARM(' DSN3EP,=DB41,S') 00014857
SUBSYS SUBNAME(DB51)        00014957
  INITRTN(DSN3INI)          00015057
  INITPARM(' DSN3EP,=DB51,S') 00015157
SUBSYS SUBNAME(DBA1)        00015257
  INITRTN(DSN3INI)          00015357
  INITPARM(' DSN3EP,=DBA1,S,DSGA') 00015457
SUBSYS SUBNAME(DBA2)        00015557
  INITRTN(DSN3INI)          00015657
```

| | |
|---|----------|
| INITPARM(' DSN3EP,=DBA2,S,DSGA') | 00015757 |
| SUBSYS SUBNAME(DBB1) | 00015857 |
| INITRTN(DSN3INI) | 00015957 |
| INITPARM(' DSN3EP,=DBB1,S,DSGB') | 00016057 |
| SUBSYS SUBNAME(DBB2) | 00016157 |
| INITRTN(DSN3INI) | 00016257 |
| INITPARM(' DSN3EP,=DBB2,S,DSGB') | 00016357 |
| SUBSYS SUBNAME(DBC1) | 00016457 |
| INITRTN(DSN3INI) | 00016557 |
| INITPARM(' DSN3EP,=DBC1,S,DSGC') | 00016657 |
| SUBSYS SUBNAME(DBC2) | 00016757 |
| INITRTN(DSN3INI) | 00016857 |
| INITPARM(' DSN3EP,=DBC2,S,DSGC') | 00016957 |
| SUBSYS SUBNAME(DBC3) | 00017057 |
| INITRTN(DSN3INI) | 00017157 |
| INITPARM(' DSN3EP,=DBC3,S,DSGC') | 00017257 |
| SUBSYS SUBNAME(DB2P) | 00017357 |
| INITRTN(DSN3INI) | 00017457 |
| INITPARM(' DSN3EPX,-DB2P,S,DBHG') | 00017557 |
| SUBSYS SUBNAME(DB2S) | 00017657 |
| INITRTN(DSN3INI) | 00017757 |
| INITPARM(' DSN3EPX,-DB2S,S,DBHG') | 00017857 |
| SUBSYS SUBNAME(DB2R) /* DB2 V5 WITH LATEST PTFS 6/98 */ | 00017957 |
| INITRTN(DSN3INI) | 00018057 |
| INITPARM(' DSN3EPX,=DB2R,S,RED') | 00018157 |
| SUBSYS SUBNAME(DB2B) /* DB2 V510 SUBSYSTEM ON SC48 */ | 00018257 |
| INITRTN(DSN3INI) | 00018357 |
| INITPARM(' DSN3EPX,=DB2B,S') | 00018457 |
| SUBSYS SUBNAME(DB2A) /* DB2 V510 DB2V510A */ | 00018557 |
| INITRTN(DSN3INI) | 00018657 |
| INITPARM(' DSN3EPX,=DB2A,S') | 00018757 |
| SUBSYS SUBNAME(DB2C) /* DB2 V510 DB2V510C */ | 00018857 |
| INITRTN(DSN3INI) | 00018957 |
| INITPARM(' DSN3EPX,=DB2C,S') | 00019057 |
| SUBSYS SUBNAME(DB2D) /* DB2 V510 DB2V510D */ | 00019157 |
| INITRTN(DSN3INI) | 00019257 |
| INITPARM(' DSN3EPX,=DB2D,S') | 00019357 |
| SUBSYS SUBNAME(DB2E) /* DB2 V510 DB2V510E */ | 00019457 |
| INITRTN(DSN3INI) | 00019557 |
| INITPARM(' DSN3EPX,=DB2E,S') | 00019657 |
| SUBSYS SUBNAME(DB2F) /* DB2 V510 DB2V510E */ | 00019757 |
| INITRTN(DSN3INI) | 00019857 |
| INITPARM(' DSN3EPX,=DB2F,S') | 00019957 |
| SUBSYS SUBNAME(DB2G) /* DB2 V510 DB2V510E */ | 00020057 |
| INITRTN(DSN3INI) | 00020157 |
| INITPARM(' DSN3EPX,=DB2G,S') | 00020257 |
| SUBSYS SUBNAME(DB2H) /* DB2 V510 DB2V510H */ | 00020357 |
| INITRTN(DSN3INI) | 00020457 |
| INITPARM(' DSN3EPX,=DB2H,S') | 00020557 |
| SUBSYS SUBNAME(DB2I) /* DB2 V510 DB2V510I */ | 00020657 |
| INITRTN(DSN3INI) | 00020757 |
| INITPARM(' DSN3EPX,=DB2I,S') | 00020857 |
| SUBSYS SUBNAME(DB2J) /* DB2 V610 DB2V610J */ | 00020957 |
| INITRTN(DSN3INI) | 00021057 |
| INITPARM(' DSN3EPX,=DB2J,S') | 00021157 |
| SUBSYS SUBNAME(DB2K) /* DB2 V510 DB2V510K */ | 00021257 |
| INITRTN(DSN3INI) | 00021357 |
| INITPARM(' DSN3EPX,=DB2K,S') | 00021457 |
| SUBSYS SUBNAME(DB2L) /* DB2 V510 DB2V510K */ | 00021557 |

| | |
|---|----------|
| INITRTN(DSN3INI) | 00021657 |
| INITPARM(' DSN3EPX,=DB2L,S') | 00021757 |
| SUBSYS SUBNAME(DB2N) /* DB2 V510 DB2V510N */ | 00021857 |
| INITRTN(DSN3INI) | 00021957 |
| INITPARM(' DSN3EPX,=DB2N,S') | 00022057 |
| SUBSYS SUBNAME(DB2V) /* DB2 V510 DB2V510N */ | 00022157 |
| INITRTN(DSN3INI) | 00022257 |
| INITPARM(' DSN3EPX,=DB2V,S') | 00022357 |
| SUBSYS SUBNAME(PSP) /* SUBSYSTEM NAME FOR BATCHPIPES */ | 00022457 |
| SUBSYS SUBNAME(BP01) /* BATCHPIPES V1R2 */ | 00022557 |
| SUBSYS SUBNAME(RACF) /* RACF SUBSYS */ | 00022657 |
| INITRTN(IRRSSIOO) | 00022757 |
| INITPARM('#,M') | 00022857 |
| SUBSYS SUBNAME(TNF) /* TCP/IP */ | 00022957 |
| /* INITRTN(MVPTSSI) */ | 00023057 |
| SUBSYS SUBNAME(VMCF) /* TCP/IP */ | 00023157 |
| /* INITRTN(MVPXSSI) */ | 00023257 |
| /* INITPARM(' WTSC&SYSCLONE') */ | 00023357 |
| SUBSYS SUBNAME(T8) /* BDT FOR JES3 */ | 00023457 |
| INITRTN(BDTSSINI) | 00023557 |
| INITPARM(' WTSCPLX3,C=/,D=Y,TQIEN=N,TQIREQ=N') | 00023657 |
| SUBSYS SUBNAME(T9) /* BDT FOR JES3 */ | 00023757 |
| INITRTN(BDTSSINI) | 00023857 |
| INITPARM(' WTSCPLX9,C=+,D=N,TQIEN=N,TQIREQ=N') | 00023957 |
| SUBSYS SUBNAME(EKGX) /* RODM */ | 00024057 |
| SUBSYS SUBNAME(AOFA) /* NETVIEW */ | 00024157 |
| SUBSYS SUBNAME(NETV) /* NETVIEW */ | 00024257 |
| SUBSYS SUBNAME(NETC) /* NETVIEW */ | 00024357 |
| SUBSYS SUBNAME(CNM) /* NETVIEW */ | 00024457 |
| SUBSYS SUBNAME(CICS) /* CICS */ | 00024557 |
| SUBSYS SUBNAME(BLSR) /* BLSR SUBSYSTEM FOR LS972106 */ | 00024957 |
| INITRTN(CSRBISUB) | 00025057 |
| SUBSYS SUBNAME(OPCT) /* TME10 OPC V2R1 TRACKER FOR LS972106 */ | 00025157 |
| INITRTN(EQQINITA) | 00025257 |
| INITPARM('400,A') | 00025357 |
| SUBSYS SUBNAME(OPCW) /* TME10 OPC V2R1 CONTROLLER FOR LS972106 */ | 00025457 |
| INITRTN(EQQINITA) | 00025557 |
| INITPARM('400,A') | 00025657 |
| SUBSYS SUBNAME(BLX1) /* INFORMATION/MANAGEMENT */ | 00050000 |
| SUBSYS SUBNAME(LOGR) /* LOGR */ | 00060000 |
| INITRTN(IXGSSINT) | 00070000 |
| SUBSYS SUBNAME(EJWS) /* (E)JES/2 */ | 00080000 |
| INITRTN(EJWSSIN) | 00090000 |
| SUBSYS SUBNAME(OAM1) /* OAM */ | 00100000 |
| INITRTN(CBRINIT) | 00110000 |
| SUBSYS SUBNAME(CSQ1) /* MQSERIES #1 */ | 00120003 |
| INITRTN(CSQ3INI) | 00130003 |
| INITPARM(' CSQ3EPX,MQS1,S') | 00140005 |
| SUBSYS SUBNAME(CSQ2) /* MQSERIES #2 */ | 00150003 |
| INITRTN(CSQ3INI) | 00160003 |
| INITPARM(' CSQ3EPX,MQS2,S') | 00170005 |
| SUBSYS SUBNAME(MQSX) | 00170160 |
| INITRTN(CSQ3INI) | 00170258 |
| INITPARM(' CSQ3EPX,=MQSX,S') | 00170360 |
| SUBSYS SUBNAME(MQSV) | 00170460 |
| INITRTN(CSQ3INI) | 00170560 |
| INITPARM(' CSQ3EPX,=MQSV,S') | 00170660 |
| SUBSYS SUBNAME(MQSY) | 00170760 |
| INITRTN(CSQ3INI) | 00170860 |

| | |
|--|----------|
| INITPARM(' CSQ3EPX,=MQSY,S') | 00170960 |
| SUBSYS SUBNAME(CQS1) /* IMS V6 COMMON QUEUE SERVER */ | 00171018 |
| SUBSYS SUBNAME(CQS2) /* IMS V6 COMMON QUEUE SERVER */ | 00172018 |
| SUBSYS SUBNAME(SBO1) /* SMARTBATCH BATCHPIPES SUBSYS */ | 00180013 |
| SUBSYS SUBNAME(ASFM) /* SMARTBATCH CONTROL SUBSYS */ | 00190013 |
| SUBSYS SUBNAME(ASFC) /* SMARTBATCH PERFORMANCE SUBSYS */ | 00200013 |
| SUBSYS SUBNAME(ASFX) /* SMARTBATCH X-JOB ENTRY SUBSYS */ | 00210013 |
| SUBSYS SUBNAME(MPM) /* ORACLE */ | 00260038 |
| SUBSYS SUBNAME(TNS4) /* ORACLE TNS4 */ | 00281052 |
| SUBSYS SUBNAME(XCMA) INITRTN(GXTINIT) /* XCM FOR HPS #1 */ | 00290041 |
| SUBSYS SUBNAME(RVA1) /* IXFP */ | 00310043 |
| INITRTN(SIBSSIPL) | 00320043 |
| INITPARM(' DYNDISR(S),INIT(N),MIH(S)') | 00330043 |
| SUBSYS SUBNAME(AXM) INITRTN(AXMSI) | 00340044 |

Appendix C. System programmers toolbox

This appendix contains sample JCL streams for many system programmer tasks. These are samples, and will need to be tailored for your installation. Updates need to reflect your *data set names*, *DASD volsers*, *device types*, and *volume addresses*.

| <i>Table 3 (Page 1 of 2). System programmer JCL toolbox</i> | |
|---|--|
| Figure | Function |
| JCL Sample 1 | DFSMSDss compress data sets on VOLSER=xxxxx |
| JCL Sample 2 | Convert DASD volume and all its data sets to SMS |
| JCL Sample 3 | DFSMSDss data set copy and re-catalog |
| JCL Sample 4 | DFSMSDss COPYDUMP JCL copy dump data set |
| JCL Sample 5 | DFSMSDss copy ALLDATA including VOLID |
| JCL Sample 6 | DFSMSDss DELETE |
| JCL Sample 7 | DFSMSDss JCL to DUMP data sets |
| JCL Sample 8 | DFSMSDss full volume physical dump |
| JCL Sample 9 | DFSMSDss job to RELEASE unused space |
| JCL Sample 10 | DFSMSDss logical data set restore |
| JCL Sample 11 | DFSMSDss full volume RESTORE from tape |
| JCL Sample 12 | DFSMSDss VSAM data set copy |
| JCL Sample 13 | IDCAMS ALTER job to RENAME a data set |
| JCL Sample 14 | AMASPZAP Job to ZAP a load module |
| JCL Sample 15 | AMBLIST job to LIST load module and CSECTs |
| JCL Sample 16 | COBOL compile/LKED JCL |
| JCL Sample 17 | IEBCOPY data set compress |
| JCL Sample 18 | IDCAMS job to define an alias |
| JCL Sample 19 | IEFBR14 define data set |
| JCL Sample 20 | IDCAMS job to define a VSAM data set |
| JCL Sample 21 | IDCAMS job to define a PAGESPACE data set |
| JCL Sample 22 | IDCAMS DELETE and DEFINE VVDS |
| JCL Sample 23 | IEFBR14 job to delete a data set |
| JCL Sample 24 | IDCAMS delete VVR |
| JCL Sample 25 | Produce error log report from Coupling Facility |
| JCL Sample 26 | Produce error log report from SYS1.LOGREC data set |
| JCL Sample 27 | ICKDSF job to perform DASD volume analysis |
| JCL Sample 28 | ICKDSF REFORMAT can be used to change the VOLSER |
| JCL Sample 29 | ICKDSF job to INITIALIZE a DASD volume |
| JCL Sample 30 | ICKDSF job to INITIALIZE a DASD volume |
| JCL Sample 31 | ICKDSF job to INSPECT a DASD device |
| JCL Sample 32 | ICFDSF job to BUILD an indexed VTOC |
| JCL Sample 33 | IEBCOPY partitioned data sets |
| JCL Sample 34 | IEBGENER to copy data sets |

| Table 3 (Page 2 of 2). System programmer JCL toolbox | |
|--|--|
| Figure | Function |
| JCL Sample 35 | IEFBR14 job to allocate a partitioned data set |
| JCL Sample 36 | IEHINITT to tape volume |
| JCL Sample 37 | IEHLIST job to list a VTOC |
| JCL Sample 38 | Assembler JCL (ASMA90) |
| JCL Sample 39 | Sample LINKEDIT job |
| JCL Sample 40 | IDCAMS job to review CACHE data for a volume |
| JCL Sample 41 | IDCAMS LISTCAT |
| JCL Sample 42 | Sample job to dump the SMF RACF records and report |
| JCL Sample 43 | IDCAMS REPRO |
| JCL Sample 44 | Stand-alone dump generation JCL |
| JCL Sample 45 | Clear SMF data set (SYS1.MANx) |
| JCL Sample 46 | Dump SMF SYS1.MANx data set |
| JCL Sample 47 | IDCAMS user catalog DISCONNECT |

```

//jobcard
//*
//*****
//*      DFSMSdss Compress Data Sets on VOLSER=xxxxx
//*      This example will compress all data sets with HLQ INSTALL
//*****
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK
//COMPVOL DD UNIT=SYSALLDA,VOL=SER=xxxxxx,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  COMPRESS INCLUDE( -
                INSTALL.*.* -
                ) -
  DDNAME(COMPVOL)

```

Figure 190. Sample 1 - DFSMSdss compress data sets on VOLSER=xxxxx

```

//jobcard
//*
//*****
//*      Convert DASD Volume and all its Data Sets to SMS Control
//*****
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK
//CONVERT DD UNIT=SYSALLDA,VOL=SER=xxxxxx,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  CONVERTV SMS DDNAME(CONVERT)
//
//*      The following statement would convert from SMS to NONSMS
  CONVERTV NOSMS DDNAME(CONVERT) TEST

```

Figure 191. Sample 2 - Convert DASD volume and all its data sets to SMS

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*   DFSMSdss Data Set Copy to New Volume and Re-Catalog 00004000
//*   This example will copy ALL data sets as identified by 00004100
//*   the DATASET(INCLUDE(**) parameter and REPLACE any    00004200
//*   existing data sets with the same name on the TARGET volume. 00004300
//*****                                00005000
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK,PARM=' UTILMSG=YES'    00010000
//SOURCE  DD  UNIT=SYSALLDA,VOL=SER=sourcevol,DISP=SHR      00020000
//TARGET  DD  UNIT=SYSALLDA,VOL=SER=targetvol,DISP=SHR     00030000
//SYSPRINT DD  SYSOUT=*                                     00040000
//SYSIN   DD  *                                           00050000
COPY     DATASET(INCLUDE( -                                00060000
                               ** -                          00070000
                               )) -                          00080000
        INDD(SOURCE) -                                     00090000
        OUTDD(TARGET) -                                    00100000
        DELETE -                                           00110000
        CATALOG -                                          00120000
        REPLACE                                           00130000

```

Figure 192. Sample 3 - DFSMSdss data set copy and re-catalog

```

//jobcard                                00001000
//*                                       00010000
//*****                                00011000
//*   DFSMSdss COPYDUMP JCL to Copy a DFSMSdss Dump Data Set 00011100
//*   from DASD to CART                                       00011100
//*****                                00012000
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK                        00020000
//DUMPIN  DD  DSN=DFSMSdss.source.data.set,DISP=SHR        00030000
//DUMPOUT DD  DSN=DFSMSdss.target.copy.data.set,           00040000
//        DISP=(NEW,CATLG),LABEL=EXPDT=99000,             00050000
//        UNIT=(CART,1,DEFER),VOL=(,.,15)                 00060000
//SYSPRINT DD  SYSOUT=*                                     00070000
//SYSIN   DD  *                                           00080000
COPYDUMP INDD(DUMPIN) -                                    00090000
        OUTDD(DUMPOUT)                                     00100000
/*                                                         00110000

```

Figure 193. Sample 4 - DFSMSdss COPYDUMP JCL copy dump data set

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//* DFSMSdss Copy ALLDATA including VOLID from one DASD vol to another 00004000
//*****                                00005000
//ADRDSSU EXEC PGM=ADRDSSU,REGION=OK    00010000
//SYSPRINT DD  SYSOUT=*                  00020000
//SYSIN  DD   *                           00030000
  COPY   INDY(sourcevol) -                00040000
         OUTDY(targetvol) -              00050000
         FULL -                           00060000
         CANCELERROR -                    00070000
         ALLDATA(*) -                     00080000
         ALLEXCP -                         00090000
         OPT(4) -                          00100000
         PURGE -                           00110000
         COPYV                              00111000

```

Figure 194. Sample 5 - DFSMS copy ALLDATA including VOLID

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//* DFSMSdss DELETE JCL will delete all cataloged data sets          00004000
//* identified in the DATASET(INCLUDE( statement)).                    00005000
//* In this example, all data sets with the HLQ OS390SMP.              00006000
//* Use with CAUTION.                                                  00007000
//*****                                00008000
//ADRDSSU EXEC PGM=ADRDSSU,REGION=OK    00010000
//TAPE   DD  DUMMY                    00020000
//SYSPRINT DD  SYSOUT=*                  00030000
//SYSIN  DD   *                           00040000
  DUMP DATASET(INCLUDE( -                00050000
                OS390SMP.** -            00060000
                )) -                      00070000
  OUTDD(TAPE) -                          00080000
  DELETE                                  00090000

```

Figure 195. Sample 6 - DFSMSdss DELETE

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*   DFSMSdss JCL to DUMP Data Sets from DASD if the DSN Change Flag 00004000
//*   is set. Reset this flag at the completion of the dump.           00005000
//*   This example excludes VSAM data sets.                             00005100
//*****                                00006000
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK    00010000
//SYSPRINT DD SYSOUT=*,HOLD=YES         00020000
//DUMP DD DSN=dump.data.set.name,DISP=(NEW,CATLG), 00030000
//      UNIT=(cart,1,DEFER),VOL=(,,15), 00040000
//      LABEL=(1,SL,EXPDT=99000)        00050000
//DASDVOL DD UNIT=3390,VOL=SER=xxxxxx,DISP=OLD 00070000
//SYSIN DD *                             00130000
  DUMP DATASET( -                         00140000
    BY((DSCHA,EQ,YES),(DSORG,NE,VSAM)) - 00150000
    INDD( -                                00160000
      DASDVOL -                            00170000
    ) -                                     00180000
    OUTDD(DUMP) -                          00190000
    RESET -                                 00200000
    TOLERATE(ENQFAILURE)                   00210000

```

Figure 196. Sample 7 - DFSMSdss JCL to DUMP data sets

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*   DFSMSdss Full Volume Physical Dump                               00004000
//*****                                00005000
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK    00010000
//SYSPRINT DD SYSOUT=*                   00020000
//DASDIN DD UNIT=SYSALLDA,VOL=SER=xxxxxx,DISP=SHR 00030000
//TAPEOUT DD DSN=tape.data.set.name,DISP=(NEW,CATLG), 00040000
//      UNIT=CART,LABEL=EXPDT=99000      00050000
//SYSIN DD *                             00060000
  DUMP INDD(DASDIN) OUTDD(TAPEOUT)       00070000
/*                                       00080000

```

Figure 197. Sample 8 - DFSMSdss full volume physical dump

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*   DFSMSdss Job to RELEASE Unused Space from eligible Data Sets
00004000
//*   In this example ALL data sets on the VOLUME identified           00004100
//*   by DDNAME(DASDVOL) will have unused space released.           00004200
//*   This does not apply to guaranteed space VSAM data sets.       00004300
//*****                                00005000
//RELEASE EXEC PGM=ADDRSSU,REGION=OK           00010000
//DASDVOL DD UNIT=SYSALLDA,VOL=SER=xxxxxx,DISP=SHR           00020000
//SYSPRINT DD SYSOUT=*                               00030000
//SYSIN DD *                                         00040000
RELEASE INCLUDE(**) DDNAME(DASDVOL)           00050000

```

Figure 198. Sample 9 - DFSMSdss job to RELEASE unused space

```

//jobcard                                00001000
//*                                       00010000
//*****                                00011000
//*   DFSMSdss Logical Data Set Restore from TAPE to DISK as
//*   identified by OUTDY(xxxxxx)
//*   This example will process ALL data sets on the dump tape.
//*   The DATASET(INCLUDE(**)) identifies all data sets.
//*****                                00014000
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK           00020000
//TAPE DD DSN=dump.data.set.name,DISP=SHR       00030000
//SYSPRINT DD SYSOUT=*                               00040000
//SYSIN DD *                                         00050000
RESTORE INDD(TAPE) -                               00060000
        DATASET(INCLUDE( -                          00070000
                ** -                                00080000
                ) -                                00090000
        ) -                                         00100000
        OUTDY(xxxxxx) -                            00110000
        CATALOG                                    00120000

```

Figure 199. Sample 10 - DFSMSdss logical data set restore

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*   DFSMSdss Full Volume RESTORE from TAPE
//*****                                00004000
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK           00010000
//SYSPRINT DD SYSOUT=*                               00020000
//TAPEIN DD DSN=tape.dump.data.set,DISP=SHR     00030000
//DASDOUT DD UNIT=SYSALLDA,VOL=SER=xxxxxx,DISP=SHR 00040000
//SYSIN DD *                                         00050000
RESTORE INDD(TAPEIN) OUTDD(DASDOUT) CANCELERROR PURGE 00060000
//                                               00070000

```

Figure 200. Sample 11 - DFSMSdss full volume RESTORE from tape


```

//jobname                                00001000
//*                                       00002000
//*****                                00003000
//*   DFSMSdss VSAM Data Set Copy and Data Component ALTER 00004000
//*****                                00005000
//ADDRSSU EXEC PGM=ADDRSSU,REGION=OK,PARM=' UTILMSG=YES' 00010000
//DASDVOL DD UNIT=SYSALLDA,VOL=SER=dasdvol,DISP=SHR      00020000
//SYSPRINT DD SYSOUT=*                                  00030000
//SYSIN DD *                                           00040000
COPY DATASET(INCLUDE( -                                00050000
                vsam.data.set.cluster.name -          00060000
                )) -                                    00070000
RENAMEU((old.vsam.data.set.cluster.name, -            00080000
        new.vsam.data.set.cluster.name)) -            00081000
OUTDD(DASDVOL) -                                       00090000
CATALOG -                                              00100000
REPLACE                                               00110000
//*                                       00120000
//*****                                00121000
//* This IDCAMS Step will Rename the VSAM Data component 00122000
//*****                                00123000
//IDCAMS EXEC PGM=IDCAMS                               00130000
//SYSPRINT DD SYSOUT=*                                  00140000
//SYSIN DD *                                           00150000
ALTER old.vsam.data.set.clustername.data -            00160000
NEWNAME(new.vsam.data.set.cluster.name.data)         00170000

```

Figure 201. Sample 12 - DFSMSdss VSAM data set copy

```

//jobcard                                00001000
//*****                                00002000
//*   IDCAMS ALTER job to RENAME a Data Set              00003000
//*****                                00004000
//IDCAMS EXEC PGM=IDCAMS                               00010000
//SYSPRINT DD SYSOUT=*                                  00020000
//SYSIN DD *                                           00030000
ALTER old.data.set.name -                              00040000
NEWNAME(new.data.set.name)                            00050000

```

Figure 202. Sample 13 - IDCAMS ALTER job to RENAME a data set

```

//jobcard                                00010000
//*                                       00020000
//*****00030000
//*      AMASPZAP Job to ZAP a Load Module 00040000
//*****00050000
//AMASPZAP EXEC PGM=AMASPZAP,REGION=4096K 00060000
//SYSPRINT DD  SYSOUT=*                   00070000
//SYSLIB  DD  DSN=load.module.data.set,DISP=SHR 00080000
//SYSIN   DD  *                               00090000
NAME loadmod csect                        00100000
*                                           00110000
VER address value                          00120000
*           - VER = Verify (Verify that the data at this 00130000
*                               address is the correct value) 00131000
REP address new_value                       00140000
*           - REP = REPLACE (If the verify is successful 00150000
*                               replace the data at the      00151000
*                               address specified with the    00152000
*                               new_value)                    00153000
CHECKSUM xxxxxxxx - CHECKSUM is optional                00160000
*           CHECKSUM is calculated by adding the          00170000
*           VERify address and value to the              00180000
*           REPplace address and value.                  00190000
* For example: VER 002C D7C7D440                       00200000
*              REP 002C C3D4C440                       00210000
*****00211000
* CHECKSUM calculation: 002CD7C7 +                      00220000
*                       D440002C +                      00230000
*                       C3D4C440 +                      00240000
*                       -----                          00250000
*           CHECKSUM   = 984193CC The CHECKSUM can be used to 00260000
*                       ----- validate the change.      00270000

```

Figure 203. Sample 14 - AMASPZAP job to ZAP a load module

```

//jobname                                00010000
//*                                       00020000
//*****00021000
//*      AMBLIST job to LIST Load Module and CSECTS 00022000
//*****00023000
//LIST EXEC PGM=AMBLIST                    00030000
//SYSPRINT DD  SYSOUT=*                   00040000
//SYSLIB  DD  DSN=load.library.data.set,DISP=SHR 00050000
//SYSIN   DD  *                               00060000
LISTLOAD MEMBER=(loadmodule)              00080000
//                                           00090000

```

Figure 204. Sample 15 - AMBLIST job to LIST load module and CSECTS

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*          COBOL Compile/LKED JCL      00003100
//*****                                00004000
//COBOL EXEC PGM=IKFCBL00,REGION=2048K,  00010000
//      PARM=' SXREF,APOST,CLIST,DMAP,ADV,OPT,SIZE(500K),BUF(200K)' 00020000
//SYSPRINT DD SYSOUT=*                   00040000
//SYSUT1 DD UNIT=VIO,SPACE=(460,(700,100)) 00050000
//SYSUT2 DD UNIT=VIO,SPACE=(460,(700,100)) 00060000
//SYSUT3 DD UNIT=VIO,SPACE=(460,(700,100)) 00070000
//SYSUT4 DD UNIT=VIO,SPACE=(460,(700,100)) 00080000
//SYSLIN DD DSN=&&LOADSET,UNIT=SYSDA,DISP=(MOD,PASS), 00090000
//      SPACE=(80,(500,100)),DCB=BLKSIZE=3200 00100000
//SYSLIB DD DSN=cobol.source.include.data.set,DISP=SHR 00120000
//SYSIN DD DSN=cobol.source.data.set(program),DISP=SHR 00130000
/*                                       00140000
//LINK EXEC PGM=HEWL,                   00150000
//      PARM=' LIST,MAP,XREF,LET',       00160000
//      COND=(8,LT,COBOL)               00170000
//SYSUT1 DD UNIT=VIO,SPACE=(TRK,(20,20)) 00180000
//SYSLIST DD SYSOUT=*                   00190000
//SYSPRINT DD SYSOUT=*                   00200000
//SYSLMOD DD DSN=loadmod.data.set,DISP=SHR 00210000
//SYSLIB DD DSN=SYS1.VSCLLIB,DISP=SHR    00220000
//      DD DSN=concatenation.data.sets,DISP=SHR 00230000
//OBJDATA DD DSN=&&LOADSET,DISP=SHR      00250000
//SYSLIN DD *                            00260000
//      INCLUDE OBJDATA                  00270000
//      ENTRY loadmodname                 00280000
//      NAME loadmodname(R)              00290000

```

Figure 205. Sample 16 - COBOL compile/LKED JCL

```

//jobcard                                00010000
//*                                       00020000
//*****                                00030000
//*          IEBCOPY Data Set Compress  00030100
//*****                                00031000
//*                                       00040000
//COMPRESS EXEC PGM=IEBCOPY,REGION=OK,PARM=COMPRESS 00060000
//SYSUT2 DD DSN=data.set.name,DISP=SHR,  00070000
//      UNIT=SYSALLDA                   00080001
//SYSPRINT DD SYSOUT=*                   00090000
//SYSIN DD DUMMY                         00100000

```

Figure 206. Sample 17 - IEBCOPY data set compress

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*   IDCAMS Job to Define an ALIAS      00004000
//*****                                00005000
//DEFALIAS EXEC PGM=IDCAMS                00010000
//SYSPRINT DD SYSOUT=*                    00020000
//SYSIN DD *                              00030000
  DEFINE ALIAS( -                          00040000
    NAME(alias) -                          00050000
    RELATE(user.catalog) -                 00060000
  ) -                                       00070000
  CATALOG(master.catalog)                  00080000
//                                          00090000

```

Figure 207. Sample 18 - IDCAMS job to define an alias

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*   IEFBR14 Define Data Set            00003100
//*****                                00004000
//IEFBR14 EXEC PGM=IEFBR14                00010000
//D1 DD DSN=data.set.name=(NEW,CATLG),    00020000
// UNIT=SYSALLDA,VOL=SER=xxxxxx,SPACE=(CYL,(20,0)), 00030000
// LRECL=80,BLKSIZE=6160                  00040000

```

Figure 208. Sample 19 - IEFBR14 define data set

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*   IDCAMS Job to Define a VSAM Data Set 00004000
//*****                                00005000
//IDCAMS EXEC PGM=IDCAMS                  00010000
//SYSPRINT DD SYSOUT=*                    00020000
//SYSIN DD *                              00030000
  DEFINE CLUSTER -                          00040000
    (NAME(cluster.data.set.name) -         00050000
    VOLUMES(xxxxxx) -                      00060000
    RECORDSIZE(313 313) -                  00070000
    INDEXED -                              00080000
    SPEED -                                00090000
    SHAREOPTIONS(3 3) -                    00100000
    KEYS(20 0)) -                           00110000
  DATA (NAME(cluster.data.set.name.DATA) - 00120000
    CYLINDERS(2)) -                          00130000
  INDEX (NAME(cluster.data.set.name.INDEX) - 00140000
    TRACKS(2))                              00150000

```

Figure 209. Sample 20 - IDCAMS job to define a VSAM data set

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*   IDCAMS Job to Define a PAGESPACE Data Set 00003100
//*****                                00004000
//IDCAMS EXEC PGM=IDCAMS                  00010000
//SYSPRINT DD SYSOUT=*                    00030000
//DASDVOL DD DISP=OLD,UNIT=3390,VOL=SER=xxxxxx 00040000
//SYSIN DD *                              00050000
    DEFINE PAGESPACE -                    00060000
        (NAME(SYS1.Vxxxxxx.LOCAL) -      00070000
         FILE(DASDVOL) -                 00080000
         CYLINDERS(3390) -               00090000
         VOLUME(xxxxxx))                 00100000
//                                         00110000

```

Figure 210. Sample 21 - IDCAMS job to define a PAGESPACE data set

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*   IDCAMS DELETE and DEFINE VVDS          00003100
//*****                                00004000
//IDCAMS EXEC PGM=IDCAMS,REGION=4096K    00010000
//DASDVOL DD UNIT=SYSALLDA,VOL=SER=xxxxxx,DISP=SHR 00020000
//SYSPRINT DD SYSOUT=*                    00030000
//SYSIN DD *                              00040000
    DELETE SYS1.VVDS.Vxxxxxx FILE(DASDVOL) 00050000
    DEFINE CLUSTER(NAME(SYS1.VVDS.Vxxxxxx) - 00070000
                 FILE(DASDVOL) -         00080000
                 VOL(xxxxxx) -           00090000
                 TRK(10 0) -             00100000
                 NIXD)                   00110000
//                                         00120000

```

Figure 211. Sample 22 - IDCAMS DELETE and DEFINE VVDS

```

//jobcard                                00010005
//*                                        00020000
//*****                                00021005
//*   IEFBR14 Job to Delete a Data Set      00022005
//*****                                00023005
//STEP1 EXEC PGM=IEFBR14                  00030000
//SYSPRINT DD SYSOUT=*                    00040002
//DELDSN DD DSN=data.set.name,           00050005
// DISP=(OLD,DELETE,DELETE)              00060005

```

Figure 212. Sample 23 - IEFBR14 job to delete a data set

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*      IDCAMS Delete VVR                00003100
//*****                                00004000
//IDCAMS  EXEC PGM=IDCAMS                 00010000
//SYSPRINT DD  SYSOUT=*                    00020000
//DASDVOL DD  UNIT=3390,VOL=SER=xxxxxx,DISP=SHR 00030000
//SYSIN   DD  *                            00040000
      DELETE data.set.nsme                -      00050000
           FILE(xxxxxx) VVR                -      00060000
           CAT(catalog.where.data.set.name.resides) 00070000

```

Figure 213. Sample 24 - IDCAMS delete VVR

```

//jobcard                                00001008
//*                                        00002000
//*****                                00002107
//*      Produce Error Log Report From Coupling Facility 00002207
//*****                                00002307
//EREPNOW EXEC PGM=IFCEREPI,REGION=4M,     00003004
//      PARM=(' HIST,ACC=N,TABSIZE=512K,PRINT=PS,TYPE=SIE') 00004006
//ACCIN   DD  DSN=SYSPLEX.LOGREC.ALLRECS,  00005004
//      DISP=SHR,                                       00006001
//      SUBSYS=(LOGR,IFBSEXIT,' FROM=(1999/125),TO=YOUNGEST', 00007106
//      ' SYSTEM=SC42'),                                00007206
//      DCB=(RECFM=VB,BLKSIZE=4000)                 00007306
//DIRECTWK DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG) 00008003
//TOURIST DD  SYSOUT=*,DCB=BLKSIZE=133        00009004
//EREPPT  DD  SYSOUT=*,DCB=BLKSIZE=133        00010004
//SYSABEND DD  SYSOUT=*                        00020004
//SYSIN   DD  DUMMY                             00030002

```

Figure 214. Sample 25 - Produce error log report from Coupling Facility

```

//jobcard                                00001002
//*                                       00002000
//*****                                00003001
//*   Produce Error Log Report from SYS1.LOGREC Data Set 00003101
//*****                                00004001
//STEP1 EXEC PGM=IFCEREP1,PARM=CARD      00010000
//SYSPRINT DD SYSOUT=*                   00020000
//SERLOG DD DSN=SYS1.LOGREC,DISP=SHR     00030000
//DIRECTWK DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG) 00040000
//EREPT DD SYSOUT=*                      00050000
//TOURIST DD SYSOUT=*                    00060000
//SYSIN DD *                              00070000
PRINT=AL                                  00080000
TYPE=S                                    00090000
ACC=N                                     00100000
HIST=N                                    00101000
TABSIZ=512K                              00110000
ENDPARM                                   00120000

```

Figure 215. Sample 26 - Produce error log report from SYS1.LOGREC data set

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*   ICKDSF Job to perform DASD Volume Analysis          00004000
//*****                                00005000
//DSF EXEC PGM=ICKDSF                    00010000
//SYSPRINT DD SYSOUT=*                   00020000
//DASDVOL DD UNIT=3390,VOL=SER=xxxxxx,DISP=OLD 00030000
//SYSIN DD *                              00040000
ANALYZE DDNAME(DASDVOL)                  00050000

```

Figure 216. Sample 27 - ICKDSF job to perform DASD volume analysis

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*   ICKDSF REFORMAT can be used to Change the Volume Serial Number 00004000
//*****                                00005000
//DSF EXEC PGM=ICKDSF                    00010000
//SYSPRINT DD SYSOUT=*                   00020000
//SYSIN DD *                              00030000
REFORMAT UNITADDRESS(xxx) VERIFY(yyyyyy) VOLID(xxxxxx) 00040000

```

Figure 217. Sample 28 - ICKDSF REFORMAT can be used to change the VOLSER

```

//jobcard 00001000
//* 00002000
//***** 00003000
//* ICKDSF Job to INITIALIZE a DASD Volume 00004002
//* Note: Change UNITADDRESS(xxx) to the DASD address to be 00005000
//* initialised. 00006000
//* VERIFY(ZZZZZZ) can be replaced with NOVERIFY if you do not 00007002
//* require VOLSER verification prior to the label change. 00008000
//* VOLID must be set to the required volser name. 00009000
//* REPLY to the WTOR message, acknowledging the request
00009100
//***** 00009200
//DSF EXEC PGM=ICKDSF 00010000
//SYSPRINT DD SYSOUT=* 00020000
//SYSIN DD * 00030000
INIT UNITADDRESS(xxx) - 00040000
VERIFY(ZZZZZZ) - 00050000
VOLID(NEWVOL) - 00060000
NOPURGE 00070000

```

Figure 218. Sample 29 - ICKDSF job to INITIALIZE a DASD volume

```

//jobcard 00001000
//* 00002000
//***** 00003000
//* ICKDSF Job to INITIALIZE a DASD Volume 00004000
//***** 00005000
//ICKDSF EXEC PGM=ICKDSF 00010000
//DASDVOL DD UNIT=SYSALLDA,VOL=SER=yyyyyy,DISP=SHR 00020000
//SYSPRINT DD SYSOUT=* 00030000
//SYSIN DD * 00040000
INIT DDNAME(DASDVOL) - 00050000
VERIFY(yyyyyy) - 00060000
PURGE - 00070000
NOVALIDATE - 00080000
NOCHECK - 00090000
VOLID(newvol) - 00100000
VTOC(2,0,75) - 00110000
INDEX(0,1,29) 00120000
// 00130000

```

Figure 219. Sample 30 - ICKDSF job to INITIALIZE a DASD volume


```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*      ICKDSF Job to INSPECT a DASD device for Surface errors 00004000
//* In this example, data will be preserved, and no alternate tracks 00005000
//* will be assigned if defective tracks are found.                00006000
//*****                                00007000
//DSF      EXEC PGM=ICKDSF                                00010000
//SYSPRINT DD  SYSOUT=*                                    00020000
//DASDVOL  DD  UNIT=3390,VOL=SER=xxxxxx,DISP=OLD            00030000
//SYSIN    DD  *                                          00040000
      INSPECT DDNAME(DASDVOL) -                            00050000
              VERIFY(xxxxxx) -                            00060000
              PRESERVE -                                   00070000
              CHECK(3) -                                   00080000
              NOASSIGN -                                   00100000
              TOLERATE(ENQFAIL)                            00110000

```

Figure 220. Sample 31 - ICKDSF job to INSPECT a DASD device

```

//jobname                                00001000
//*                                        00002000
//*****                                00003000
//*      ICKDSF Job to BUILD an Indexed VTOC on DASD volume xxxxxx 00003102
//* Use BUILDIX DDNAME(DASDVOL) OSVTOC to convert an Indexed      00003202
//* VTOC to an OS VTOC.                                          00003302
//*****                                00004000
//DSF      EXEC PGM=ICKDSF                                00010000
//SYSPRINT DD  SYSOUT=*                                    00020000
//DASDVOL  DD  UNIT=SYSALLDA,VOL=SER=xxxxxx,DISP=(OLD)        00030000
//SYSIN    DD  *                                          00040000
      BUILDIX DDNAME(DASDVOL) IXVTOC                       00050000

```

Figure 221. Sample 32 - ICKDSF job to BUILD an indexed VTOC

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*      IEBCOPY Partitioned Data Sets                            00004000
//*****                                00005000
//IEBCOPY  EXEC PGM=IEBCOPY,REGION=4096K                    00010000
//SYSUT1   DD  DSN=input.partitioned.data.set,DISP=SHR        00020000
//SYSUT2   DD  DSN=output.psrtitioned.data.set,DISP=(NEW,CATLG), 00030000
//          UNIT=3390,VOL=SER=xxxxxx,SPACE=(CYL,(1,1,45)),    00040000
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)              00050000
//SYSPRINT DD  SYSOUT=*                                    00060000
//SYSIN    DD  *                                          00070000
      COPY OUTDD=SYSUT2,INDD=((SYSUT1,R))                  00080000

```

Figure 222. Sample 33 - IEBCOPY partitioned data sets

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*      IEBGENER to Copy Data Sets      00003100
//*****                                00004000
//IEBGENER EXEC PGM=IEBGENER,REGION=4096K 00010000
//SYSPRINT DD SYSOUT=*                    00020000
//SYSUT1  DD DSN=input.data.set.name,DISP=SHR 00030000
//SYSUT2  DD DSN=output.data.set.name,DISP=OLD 00040000
//SYSIN   DD DUMMY                          00050000

```

Figure 223. Sample 34 - IEBGENER to copy data sets

```

//jobcard                                00010000
//*                                        00020000
//*****                                00021000
//*      IEFBR14 Job to Allocate a Partitioned Data Set 00022000
//*****                                00023000
//STEP1   EXEC PGM=IEFBR14                00024000
//SYSPRINT DD SYSOUT=*                    00025000
//ALLOC   DD DSN=data.set.name,           00026000
//        DISP=(NEW,CATLG,DELETE),       00027000
//        UNIT=SYSDA,                     00028000
//        SPACE=(CYL,(1,0,45),           00029000
//        LRECL=80                         00030000

```

Figure 224. Sample 35 - IEFBR14 job to allocate a partitioned data set

```

//jobcard                                00001000
//*                                        00002000
//*****                                00003000
//*      IEHINITT to TAPE Volume          00003100
//*****                                00004000
//IEHINITT EXEC PGM=IEHINITT              00010000
//SYSPRINT DD SYSOUT=*                    00020000
//LABEL   DD UNIT=(3490,1,DEFER),VOL=SER=xxxxxx 00030000
//SYSIN   DD *                             00040000
LABEL INITT SER=xxxxxx,NUMBTAPE=001 OWNER=optional 00050000

```

Figure 225. Sample 36 - IEHINITT to tape volume

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*      IEHLIST job to List a VTOC      00003100
//*****                                00004000
//IEHLIST EXEC PGM=IEHLIST              00010000
//SYSPRINT DD SYSOUT=*                   00020000
//DASDVOL DD VOL=SER=xxxxxx,UNIT=SYSALLDA,DISP=OLD 00030000
//SYSIN DD *                             00040000
LISTVTOC VOL=SYSALLDA=xxxxxx           00050000

```

Figure 226. Sample 37 - IEHLIST job to list a VTOC

```

//jobcard                                00010000
//*                                       00030000
//*****                                00031000
//*      Assembler JCL - ASMA90         00031100
//*****                                00032000
//STEP1 EXEC PGM=ASMA90,                00040000
//      PARM='LIST,NOOBJECT,DECK,SYSPARM(LOCAL)'  
//                                       00050000
//SYSPRINT DD SYSOUT=*                   00060000
//SYSLIB DD DSN=syslib.data.set,DISP=SHR 00070000
//      DD DSN=syslib.concat.data.set,disp=shr 00071000
//SYSUT1 DD DSN=&&ASM1,UNIT=VIO,SPACE=(CYL,(5,5)) 00080000
//SYSUT2 DD DSN=&&ASM2,UNIT=VIO,SPACE=(CYL,(5,5)) 00090000
//SYSUT3 DD DSN=&&ASM3,UNIT=VIO,SPACE=(CYL,(5,5)) 00100000
//SYSLIN DD DSN=&&OBJECT,DISP=(NEW,PASS), 00110000
//      UNIT=SYSDA,SPACE=(CYL,(1,1)) 00120000
//SYSPUNCH DD DSN=syspunch.data.set,DISP=SHR 00130000
//SYSIN DD DSN=assembler.source.data.set(member),DISP=SHR 00140000
//*                                       00150000

```

Figure 227. Sample 38 - Assembler JCL (ASMA90)

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*      Sample LINKEDIT Job            00003100
//*****                                00004000
//LKED EXEC PGM=IEWL,PARM='AMODE(31),RMODE(ANY),AC=1'  
//SYSPRINT DD SYSOUT=*                   00020000
//LINKLIB DD DSN=include.data.set,DISP=SHR 00030000
//SYSLMOD DD DSN=target.load.library,DISP=SHR 00040000
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(5,5)) 00050000
//SYSLIN DD *                             00060000
ENTRY modulename                        00070000
INCLUDE LINKLIB(include module)         00080000
NAME loadmodname(R)                     00090000

```

Figure 228. Sample 39 - Sample LINKEDIT job

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*      IDCAMS Job to Review CACHE data for a volume 00004000
//*****                                00005000
//IDCAMS  EXEC PGM=IDCAMS                00010000
//CACHE1 DD UNIT=3390,VOL=SER=xxxxxx,DISP=SHR 00020000
//SYSPRINT DD SYSOUT=A,HOLD=YES          00030000
//SYSIN   DD *                            00040000
LISTDATA -                               00050000
COUNTS -                                 00060000
UNIT(3390) -                              00070000
VOL(xxxxxx) -                             00080000
ALL -                                       00090000
LEGEND                                     00100000

```

Figure 229. Sample 40 - IDCAMS job to review CACHE data for a volume

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*      IDCAMS LISTCAT                    00003100
//*****                                00004000
//LISTCAT EXEC PGM=IDCAMS                00010000
//SYSPRINT DD SYSOUT=A,HOLD=YES          00020000
//SYSIN   DD *                            00030000
LISTC CAT(catalog.name)                 00040000
LISTC ENT(data.set.name)                 00041000
LISTC LVL(HLQ)                           00042000
//*                                       00050000

```

Figure 230. Sample 41 - IDCAMS LISTCAT

```

//jobcard 00001000
//* 00002000
//***** 00003000
//* Sample Job to Dump the SMF RACF records an produce a Report 00003100
//***** 00004000
//IFASMFDP EXEC PGM=IFASMFDP 00010000
//SYSPRINT DD SYSOUT=* 00020000
//SYSUDUMP DD SYSOUT=* 00030000
//INDD1 DD DSN=SYS1.MAN1,DISP=SHR 00040000
//OUTDD1 DD DSN=MAN1,DISP=(NEW,PASS), 00050000
// UNIT=SYSDA,SPACE=(CYL,(1,1)) 00060000
//SYSIN DD * 00070000
INDD(INDD1,OPTIONS(DUMP)) 00080000
OUTDD(OUTDD1,TYPE(80,81)) 00090000
/* 00100000
//RACFRW EXEC PGM=IKJEFT01 00110000
//SYSTSPRT DD SYSOUT=*,HOLD=YES 00120000
//SYSOUT DD SYSOUT=*,HOLD=YES 00130000
//SYSPRINT DD SYSOUT=*,HOLD=YES 00140000
//RSMFIN DD DSN=MAN1,DISP=SHR 00150000
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR 00160000
//SORTIN DD UNIT=SYSDA,SPACE=(CYL,(5,5)) 00170000
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(5,5))
//SYSTSIN DD *
RACFRW TITLE(' RACF SMF REPORT')
SELECT DATE(99001:99365)
EVENT ALLCOMMAND
EVENT ALLSVC
LIST SORT(DATE TIME)
END

```

Figure 231. Sample 42 - Sample job to dump the SMF RACF records and report

```

//jobcard 00001000
//* 00002000
//***** 00003000
//* IDCAMS REPRO 00003100
//***** 00004000
//REPRO EXEC PGM=IDCAMS,REGION=OK 00010000
//SYSPRINT DD SYSOUT=* 00020000
//INPUT DD DSN=input.vsam.data.set.name,DISP=SHR 00030000
//OUTPUT DD DSN=output.vsam.data.set.name,DISP=SHR 00040000
//SYSIN DD * 00050000
REPRO IFILE(INPUT) - 00060000
OFFILE(OUTPUT) REPLACE 00070000
//* 00080000

```

Figure 232. Sample 43 - IDCAMS REPRO

```

//jobcard                                00010000
//*                                       00030000
//*****                                00031000
//*          Stand-alone Dump Generation JCL          00032000
//*                                       00033000
//* Note: 1. Specify correct VOLSER, Console addresses and GENPRINT 00040000
//*          data set name.                                00040100
//*          2. Delete existing SYS1.PAGEDUMP before running this JCL. 00041000
//*          3. Reply to outstanding WTOR messages.        00050000
//*                                       00060000
//*****                                00061000
//OSG      EXEC PGM=AMDSAOSG,REGION=4M                00070000
//SYSLIB   DD DSN=SYS1.MACLIB,DISP=SHR                00080000
//          DD DSN=SYS1.MODGEN,DISP=SHR              00090000
//DPLTEXT  DD DSN=SYS1.NUCLEUS(AMDSADPL),DISP=SHR     00100000
//IPLTEXT  DD DSN=SYS1.NUCLEUS(AMDSAIPD),DISP=SHR     00110000
//PGETEXT  DD DSN=SYS1.NUCLEUS(AMDSAPGE),DISP=SHR     00120000
//GENPRINT DD DSN=genprint.data.set.name,DISP=(NEW,CATLG), 00130000
//          UNIT=SYSALLDA,SPACE=(CYL,(1,1)),          00140000
//          DCB=(RECFM=FB,LRECL=133,BLKSIZE=23408)    00150000
//GENPARMS DD *                                       00160000
//          AMDSADMP IPL=D3390,VOLSER=xxxxxx,MINASID=ALL,  +00170000
//          CONSOLE=((xxx,3278),(xxx,3278),(xxx,3278)), +00180000
//          OUTPUT=TD20                                00190001
//          END                                        00200001
//*                                       00210001
//*                                       00220001

```

Figure 233. Sample 44 - Stand-alone dump generation JCL

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*          Clear SMF Data Set (SYS1.MANx)          00003100
//*****                                00004000
//SMFCLR   EXEC PGM=IFASMFDP                          00010000
//SYSPRINT DD SYSOUT=*                                00030000
//SYSUDUMP DD SYSOUT=*                                00040000
//INDD1    DD DSN=SYS1.MANx,DISP=SHR                 00050000
//SYSIN    DD *                                       00060000
//          INDD(INDD1,OPTIONS(CLEAR))                00070000

```

Figure 234. Sample 45 - Clear SMF data set (SYS1.MANx)

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*      Dump SMF SYS1.MANx Data Set      00003100
//*****                                00004000
//SMFDP   EXEC PGM=IFASMFDP              00010000
//SYSPRINT DD SYSOUT=*                   00020000
//SYSUDUMP DD SYSOUT=*                   00030000
//INDD1   DD DSN=SYS1.MANx,DISP=SHR      00040000
//OUTDD1  DD DSN=smf.manx.dump.data.set.name, 00050000
//          DISP=(NEW,CATLG,DELETE),      00060000
//          UNIT=SYSDA,                    00070000
//          SPACE=(CYL,(10,10),RLSE)     00080000
//SYSIN   DD *                             00090000
INDD(INDD1,OPTIONS(DUMP))                00100000
OUTDD(OUTDD1,TYPE(smf record types to be dumped,0:255,for ALL)) 00110000
//*                                       00120000

```

Figure 235. Sample 46 - Dump SMF SYS1.MANx data set

```

//jobcard                                00001000
//*                                       00002000
//*****                                00003000
//*      IDCAMS User Catalog DISCONNECT  00003100
//*****                                00004000
//UCATDISC EXEC PGM=IDCAMS              00010000
//SYSPRINT DD SYSOUT=*                   00030000
//SYSIN   DD *                             00060000
EXPORT user.catalog.name -               00070000
DISCONNECT -                             00080000
CATALOG(master.catalog.name)            00090000

```

Figure 236. Sample 47 - IDCAMS user catalog DISCONNECT

Appendix D. Special Notices

This publication is intended to help new system programmers who need to understand S/390 and the OS/390 operating system. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/390 Versions. See the PUBLICATIONS section of the IBM Programming Announcement for OS/390 Version 2 Release 8, Program Number 5647-A01 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

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, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM
AFP
CICS
CICS/MVS

Advanced Function Printing
AnyNet
CICS/ESA
DB2

| | |
|----------------------------------|----------------------|
| DFSMS | DFSMS/MVS |
| DFSMSdfp | DFSMSdss |
| DFSMSshsm | DFSMSrmm |
| DFSORT | ESCON |
| FICON | IBM |
| IMS | InfoPrint |
| Intelligent Printer Data Stream | IPDS |
| IP PrintWay | Language Environment |
| MVS (block letters) | MVS/DFP |
| NetSpool | NetView |
| OpenEdition | OS/390 |
| Parallel Sysplex | PrintWay |
| RACF | RAMAC |
| RMF | S/390 |
| S/390 Parallel Enterprise Server | Sysplex Timer |
| VTAM | |

The following terms are trademarks of other companies:

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjobenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

SET, SET Secure Electronic Transaction, and the SET logo are trademarks owned by Secure Electronic Transaction LLC.

UNIX is a registered trademark in the United States and other countries licensed exclusively through the Open Group.

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

Appendix E. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 IBM Redbooks

For information on ordering these ITSO publications see “How to get IBM Redbooks” on page 309.

- *OS/390 Release 5 Implementation*, SG24-5151
- *OS/390 Release 4 Implementation*, SG24-2089
- *OS/390 Release 3 Implementation*, SG24-2067
- *OS/390 Release 2 Implementation*, SG24-4834
- *cit.OS/390 Security Server 1999 Updates Technical Presentation Guide*, SG24-5627
- *Security in OS/390-based TCP/IP Networks*, SG24-5383
- *Hierarchical File System Usage Guide*, SG24-5482
- *Enhanced Catalog Sharing and Management*, SG24-5594
- *Integrated Catalog Facility Backup and Recovery*, SG24-5644
- *OS/390 Version 2 Release 6 UNIX System Services Implementation and Customization*, SG24-5178
- *IBM S/390 FICON Implementation Guide*, SG24-5169
- *Exploiting S/390 Hardware Cryptography with Trusted Key Entry*, SG24-5455
- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- *Introduction to Storage Area Network SAN*, SG2-45470
- *TCP/IP in a Sysplex*, SG24-5235
- *SecureWay Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements*, SG24-5631
- *OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 1: Configuration and Routing*, SG24-5227
- *OS/390 eNetwork Communications Server V2R7 TCP/IP Implementation Guide Volume 3: MVS Applications*, SG24-5229
- *OS/390 Workload Manager Implementation and Exploitation*, SG24-5326
- *ADSM Server-to-Server Implementation and Operation*, SG24-5244
- *Stay Cool on OS/390: Installing Firewall Technology*, SG24-2046
- *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy*, SG24-5268
- *TCP/IP OpenEdition Implementation Guide*, SG24-2141
- *IMS/ESA Version 5 Performance Guide*, SG24-4637
- *Parallel Sysplex Configuration: Overview*, SG24-2075
- *Parallel Sysplex Configuration: Cookbook*, SG24-2076
- *Parallel Sysplex Config.: Connectivity*, SG24-2077
- *DFSMS Optimizer Presentation Guide Update*, SG24-4477
- *MVS Parallel Sysplex Capacity Planning*, SG24-4680

- *Getting the Most Out of a Parallel Sysplex*, SG24-2073
- *OS/390 eNetwork Communication Server TCP/IP Implementation Guide Volume 2*, SG24-5228

E.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at <http://www.redbooks.ibm.com/> for information about all the CD-ROMs offered, updates and formats.

| CD-ROM Title | Collection Kit Number |
|--|-----------------------|
| System/390 Redbooks Collection | SK2T-2177 |
| Networking and Systems Management Redbooks Collection | SK2T-6022 |
| Transaction Processing and Data Management Redbooks Collection | SK2T-8038 |
| Lotus Redbooks Collection | SK2T-8039 |
| Tivoli Redbooks Collection | SK2T-8044 |
| AS/400 Redbooks Collection | SK2T-2849 |
| Netfinity Hardware and Software Redbooks Collection | SK2T-8046 |
| RS/6000 Redbooks Collection (BkMgr Format) | SK2T-8040 |
| RS/6000 Redbooks Collection (PDF Format) | SK2T-8043 |
| Application Development Redbooks Collection | SK2T-8037 |
| IBM Enterprise Storage and Systems Management Solutions | SK3T-3694 |

E.3 Other resources

These publications are also relevant as further information sources:

- *OS/390 Initialization and Tuning Guide*, SC28-1751
- *OS/390 Initialization and Tuning Reference*, SC28-1752
- *OS/390 Introduction and Release Guide*, GC28-1725
- *OS/390 MVS JCL User's Guide*, SC28-1758
- *OS/390 MVS JCL Reference*, GC28-1757
- *OS/390 MVS System Diagnosis: Tools and Service Aids*, LY28-1085, (available to IBM licensed customers only)
- *Interactive System Productivity Facility Getting Started*, SC34-4440
- *OS/390 Security Server (RACF) System Programmer's Guide*, SC28-1913
- *OS/390 TSO/E Customization*, SC28-1965
- *OS/390 TSO/E Primer*, GC28-1967
- *OS/390 TSO/E User's Guide*, SC28-1968
- *OS/390 SMP/E Reference*, SC28-1806
- *OS/390 SMP/E User's Guide*, SC28-1740
- *OS/390 SMP/E Commands*, SC28-1805
- *Standard Packaging Rules for MVS-Based Products*, SC23-3695
- *OS/390 MVS System Commands*, GC28-1781
- *OS/390 MVS IPCS Commands*, GC28-1754
- *OS/390 MVS IPCS User's Guide*, GC28-1756
- *DFSMS/MVS Using Data Sets*, SC26-4922
- *OS/390 Planning for Installation*, GC28-1726
- *OS/390 MVS System Data Sets Definition*, GC28-1782

- *ICKDSF R16 Refresh, User's Guide*, GC35-0033
- *OS/390 MVS System Management Facilities (SMF)*, GC28-1783
- *EREP V3R5 Reference*, GC35-0152
- *OS/390 JES2 Commands*, GC28-1790
- *OS/390 Hardware Configuration Definition User's Guide*, SC28-1848
- *DFSMS/MVS DFSMSdss Storage Administration Reference*, SC26-4929
- *IBM ServerPac for OS/390 Using the Installation Dialog*, SC28-1244
- *OS/390 Hardware Configuration Definition Planning*, GC28-1750
- *OS/390 MVS Using the Subsystem Interface*, SC28-1502
- *DFSMS/MVS Version 1 Release 4: Managing Catalogs*, SC26-4914
- *DFSMS/MVS Version 1 Release4: Access Method Services for Integrated Catalog Facility*, SC26-4906
- *DFSMS/MVS: DFSMSshm Implementation and Customization Guide*, SH21-1078
- *DFSMS/MVS Access Method Services for ICF Catalogs*, SC26-4500
- *DFSMS/MVS DFSMSdfp Storage Administration Reference*, SC26-4920
- *OS/390 eNetwork Communications Server: SNA Resource Definition Reference*, SC31-8565
- *OS/390 eNetwork Communications Server SNA Resource Definition Samples*, SC31-8566
- *OS/390 eNetwork Communications Server: SNA Operation*, SC31-8567
- *OS/390 V2R7.0 eNetwork CS IP Configuration*, SC31-8513
- *eNetwork Communications Server: IP User's Guide* GC31-8514
- *OS/390 UNIX System Services Planning*, SC28-1890
- *OS/390 TCP/IP OpenEdition: Configuration Guide* SC31-8304
- *OS/390 Open Systems Adapter Support Facility User's Guide*, SC28-1855.
- *OS/390 V2R6.0 MVS Planning: APPC/MVS Management*, GC28-1807
- *Print Services Facility for OS/390: Customization*, S544-5622
- *DFSMS/MVS Planning for Installation*, SC26-4919
- *DFSMS/MVS Implementing System-Managed Storage*, SC26-3123
- *DFSMS/MVS Remote Copy Administrator's Guide and Reference*, SC35-0169
- *DFSMS/MVS Macro Instructions for Data Sets*, SC26-4913
- *DFSMS/MVS DFSMSdfp Diagnosis Guide*, SY27-9605
- *DFSMS/MVS DFSMSdfp Advanced Services*, SC26-4921
- *DFSMS/MVS Using Magnetic Tapes*, SC26-4923
- *DFSMS/MVS Utilities*, SC26-4926
- *Service Level Reporter User's Guide: Reporting*, SH19-6530
- *DFSMS/MVS Object Access Method Application Programmer's Reference*, SC26-4917
- *DFSMS/MVS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*, SC26-4918
- *DFSORT Installation and Customization*, SC26-7041
- *DFSORT Getting Started with DFSORT R14*, SC26-4109
- *DFSMS/MVS Network File System Customization and Operation*, SC26-7029

- *DFSMS Optimizer User's Guide and Reference*, SC26-7047
- *DFSMS/MVS DFSMSdss Storage Administration Guide*, SC26-4930
- *DFSMSShsm Storage Administration Guide*, SH21-1076
- *DFSMSShsm Storage Administration Reference*, SH21-1075
- *DFSMS/MVS Network File System User's Guide*, SC26-7028
- *DFSMS/MVS DFSMSrmm Guide and Reference*, SC26-4931
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*, SC26-4932
- *MVS/ESA Storage Management Library Managing Data*, SC26-3124
- *MVS/ESA Storage Management Library Managing Storage Groups*, SC26-3125
- *MVS/ESA Storage Management Library Leading a Storage Administration Group*, SC26-3126.
- *DFSMS/MVS Using the Interactive Storage Management Facility*, SC26-4911
- *ADSTAR Distributed Storage Manager for MVS Administrator's Guide*, GC35-0277
- *OS/390 MVS Programming: Assembler Services Guide*, GC28-1762
- *OS/390 MVS Programming: Resource Recovery*, GC28-1739
- *OS/390 MVS Setting Up a Sysplex*, GC28-1779
- *OS/390 MVS Sysplex Services Guide*, GC28-1771
- *OS/390 Parallel Sysplex Systems Management*, GC28-1861
- *OS/390 MVS Systems Codes*, GC28-1780
- *OS/390 MVS System Messages Volume 1*, GC28-1784
- *OS/390 MVS System Messages Volume 2*, GC28-1785
- *OS/390 MVS System Messages Volume 3*, GC28-1786
- *OS/390 MVS System Messages Volume 4*, GC28-1787
- *OS/390 MVS System Messages Volume 5*, GC28-1788
- *OS/390 MVS Installation Exits*, SC28-1753
- *OS/390 MVS Diagnosis Reference*, SY28-1084
- *CICS User's Handbook*, SX33-1188
- *CICS Diagnosis Guide*, LX33-6093
- *MQSeries for MVS/ESA Messages and Codes*, GC33-0819

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** <http://www.redbooks.ibm.com/>

Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders by e-mail including information from the redbook fax order form to:

In United States:
Outside North America:

e-mail address: usib6fpl@ibmmail.com
Contact information is in the "How to Order" section at this site:
<http://www.elink.ibmmlink.ibm.com/pbl/pbl/>

- **Telephone Orders**

United States (toll free)
Canada (toll free)
Outside North America

1-800-879-2755
1-800-IBM-4YOU
Country coordinator phone number is in the "How to Order" section at this site:
<http://www.elink.ibmmlink.ibm.com/pbl/pbl/>

- **Fax Orders**

United States (toll free)
Canada
Outside North America

1-800-445-9269
1-403-267-4455
Fax phone number is in the "How to Order" section at this site:
<http://www.elink.ibmmlink.ibm.com/pbl/pbl/>

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at <http://w3.ibm.com/> for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

| Title | Order Number | Quantity |
|-------|--------------|----------|
| _____ | _____ | _____ |
| _____ | _____ | _____ |
| _____ | _____ | _____ |
| _____ | _____ | _____ |

First name _____ Last name _____

Company _____

Address _____

City _____ Postal code _____ Country _____

Telephone number _____ Telefax number _____ VAT number _____

- Invoice to customer number _____
- Credit card number _____

Credit card expiration date _____ Card issued to _____ Signature _____

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Glossary

A

abend. Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task executing.

ACB. Access method control block.

access. A specific type of interaction between a subject and an object that results in the flow of information from one to the other.

access authority. An authority that relates to a request for a type of access to protected resources. In RACF, the access authorities are NONE, READ, UPDATE, ALTER, and EXECUTE.

access list. A list within a profile of all authorized users and their access authorities.

access method control block (ACB). A control block that links an application program to VTAM.

ACDS. Active control data set.

ACF/VTAM. An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability. VTAM runs under MVS (OS/VS1 and OS/VS2), VSE, and VM/SP and supports direct control application programs and subsystems such as VSE/POWER.

ACIF. (1) AFP conversion and indexing facility. (2) A PSF utility program that converts a print file into AFP, MO:DCA-P, creates an index file for input data, and collects resources used by an AFP document into separate file.

action message retention facility (AMRF). A facility that, when active, retains all action messages except those specified by the installation in the MPFLSTxx member in effect.

action message sequence number. A decimal number assigned to action messages.

Advanced Function Presentation (AFP). A set of licensed programs, together with user applications, that use the all-points-addressable concept to print on presentation devices. AFP includes creating, formatting, archiving, retrieving, viewing, distributing, and printing information.

Advanced Program-to-Program Communications (APPC). A set of inter-program communication services that support cooperative transaction processing in a SNA network.

AFP. Advanced Function Presentation.

AFP Printer Driver for Windows. A component of Infoprint Server for OS/390 that runs on a Windows 95 or Windows NT workstation and creates output in AFP format, for printing on AFP printers.

AFP Viewer plug-in for Windows. A component of Infoprint Server for OS/390 that runs on a Windows 95 or Windows NT workstation and allows you to view files in AFP format.

AIX operating system. IBM's implementation of the UNIX operating system. The RS/6000 system, among others, runs the AIX operating system.

allocate. To assign a resource for use in performing a specific task.

alphanumeric character. A letter or a number.

amode. Addressing mode. A program attribute that can be specified (or defaulted) for each CSECT, load module, and load module alias. AMODE states the addressing mode that is expected to be in effect when the program is entered.

AMRF. action message retention facility

AOR. Application-owning region

APPC. Advanced Program-to-Program Communications

APPN. Advanced Peer-to-Peer Networking.

ASCII (American Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8-bit 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.

audit. To review and examine the activities of a data processing system mainly to test the adequacy and effectiveness of procedures for data security and data accuracy.

authority. The right to access objects, resources, or functions.

authorization checking. The action of determining whether a user is permitted access to a RACF-protected resource.

Authorized Program Analysis Report (APAR). A request for correction of problem caused by a defect in a current unaltered release of a program.

authorized program facility (APF)

authorized program facility (APF). A facility that permits identification of programs authorized to use restricted functions.

automated operations. Automated procedures to replace or simplify actions of operators in both systems and network operations.

AVR. Automatic volume recognition.

B

banner page. A page printed before the data set is printed.

basic mode. A central processor mode that does not use logical partitioning. Contrast with logically partitioned (LPAR) mode.

batch message processing (BMP) program. An IMS batch processing program that has access to online databases and message queues. BMPs run online, but like programs in a batch environment, they are started with job control language (JCL).

batch-oriented BMP program. A BMP program that has access to online databases and message queues while performing batch-type processing. A batch-oriented BMP does not access the IMS message queues for input or output. It can access online databases, GSAM databases, and MVS files for both input and output.

BMP. Batch message processing (BMP) program.

broadcast. (1) Transmission of the same data to all destinations. (2) Simultaneous transmission of data to more than one destination.

binary data. (1) Any data not intended for direct human reading. Binary data may contain unprintable characters, outside the range of text characters. (2) A type of data consisting of numeric values stored in bit patterns of 0s and 1s. Binary data can cause a large number to be placed in a smaller space of storage.

BIND. In SNA, a request to activate a session between two logical units (LUs).

buffer. A portion of storage used to hold input or output data temporarily.

buffered device. A device where the data is written to a hardware buffer in the device before it is placed on the paper (for example, IBM 3820).

burst. To separate continuous-forms paper into single sheets.

C

cache structure. A coupling facility structure that enables high-performance sharing of cached data by multisystem applications in a sysplex. Applications can use a cache structure to implement several different types of caching systems, including a store-through or a store-in cache.

carriage control character. An optional character in an input data record that specifies a write, space, or skip operation.

carriage return (CR). (1) A keystroke generally indicating the end of a command line. (2) In text data, the action that indicates to continue printing at the left margin of the next line. (3) A character that will cause printing to start at the beginning of the same physical line in which the carriage return occurred.

CART. Command and response token.

case-sensitive. Pertaining to the ability to distinguish between uppercase and lowercase letters.

catalog. (1) A directory of files and libraries, with reference to their locations. (2) To enter information about a file or a library into a (3) The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set.

CBPDO. Custom Built Product Delivery Offering.

CEC. Synonym for central processor complex (CPC).

central processor (CP). The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load, and other machine operations.

central processor complex (CPC). A physical collection of hardware that includes main storage, one or more central processors, timers, and channels.

CFRM. Coupling facility resource management.

channel-to-channel (CTC). Refers to the communication (transfer of data between programs on opposite sides of a channel-to-channel adapter (CTCA

channel-to-channel adapter (CTCA). An input/output device that is used a program in one system to communicate with a program in another system.

checkpoint. (1) A place in a routine where a check, or a recording of data for restart purposes, is performed. (2) A point at which information about the status of a job and the system can be recorded so that the job step can be restarted later.

checkpoint write. Any write to the checkpoint data set. A general term for the primary, intermediate, and final writes that update any checkpoint data set.

CICS. Customer Information Control System.

CICSplex. A group of connected CICS regions.

CICSplex SM. CICSplex System Manager

client. A functional unit that receives shared services from a server. See also client-server.

client-server. In TCP/IP, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

CMOS. Complementary metal-oxide semiconductor.

CNGRPxx. The SYS1.PARMLIB member that defines console groups for the system or sysplex.

code page. (1) A table showing codes assigned to character sets. (2) An assignment of graphic characters and control function meanings to all code points. (3) Arrays of code points representing characters that establish ordinal sequence (numeric order) of characters. (4) A particular assignment of hexadecimal identifiers to graphic elements.

code point. A 1-byte code representing one of 256 potential characters.

coexistence. Two or more systems at different levels (for example, software, service or operational levels) that share resources. Coexistence includes the ability of a system to respond in the following ways to a new function that was introduced on another system with which it shares resources: ignore a new function, terminate gracefully, support a new function.

command and response token (CART). A parameter on WTO, WTOR, MGCRC, and certain TSO/E commands and REXX execs that allows you to link commands and their associated message responses.

command prefix facility (CPF). An MVS facility that allows you to define and control subsystem and other command prefixes for use in a sysplex.

COMMDS. Communications data set.

complementary metal-oxide semiconductor (CMOS). A technology that combines the electrical properties of positive and negative voltage requirements to use considerably less power than other types of semiconductors.

connection. In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In Internet communications, a connection extends from a TCP application on one system to a TCP application on another system.

console. That part of a computer used for communication between the operator or user and the computer.

console group. In MVS, a group of consoles defined in CNGRPxx, each of whose members can serve as an alternate console in console or hardcopy recovery or as a console to display synchronous messages.

CONSOLxx. The SYS1.PARMLIB member used to define message handling, command processing, and MCS consoles.

control unit. Synonymous with device control unit.

conversation. A logical connection between two programs over an LU type 6.2 session that allows them to communicate with each other while processing a transaction.

conversational. Pertaining to a program or a system that carries on a dialog with a terminal user, alternately accepting input and then responding to the input quickly enough for the user to maintain a train of thought.

copy group. One or more copies of a page of paper. Each copy can have modifications, such as text suppression, page position, forms flash, and overlays.

couple data set. A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the MVS systems in a sysplex. See also sysplex couple data set.

coupling facility. A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

coupling facility channel. A high bandwidth fiber optic channel that provides the high-speed connectivity required for data sharing between a coupling facility and the central processor complexes directly attached to it.

coupling services. In a sysplex, the functions of XCF that transfer data and status between members of a group residing on one or more MVS systems in the sysplex.

CP. Central processor.

CPC. Central processor complex.

CPF. Command prefix facility.

cross-system coupling facility (XCF). XCF is a component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.

cryptography. The transformation of data to conceal its meaning.

cryptographic key

cryptographic key. A parameter that determines cryptographic transformations between plaintext and ciphertext.

CTC. Channel-to-channel.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D

DAE. Dump analysis and elimination.

daemon. A program that runs unattended to perform a standard service.

DASD. Direct access storage device.

data definition name. The name of a data definition (DD) statement, which corresponds to a data control block that contains the same name. Abbreviated as ddname.

data definition (DD) statement. A job control statement that describes a data set associated with a particular job step.

data integrity. The condition that exists as long as accidental or intentional destruction, alteration, or loss of data does not occur.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data set label. (1) A collection of information that describes the attributes of a data set and is normally stored on the same volume as the data set. (2) A general term for data set control blocks and tape data set labels.

data set separator pages. Those pages of printed output that delimit data sets.

data sharing. The ability of concurrent subsystems (such as DB2 or IMS DB) or application programs to directly access and change the same data while maintaining data integrity.

data stream. (1) All information (data and control commands) sent over a data link usually in a single read or write operation. (2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

DBCS. Double-byte character set.

DBCTL. IMS Database Control.

DBRC. Database Recovery Control.

DB2. DATABASE 2 for MVS/ESA.

DB2 data sharing group. A collection of one or more concurrent DB2 subsystems that directly access and change the same data while maintaining data integrity.

DB2 PM. DB2 Performance Monitor.

deallocate. To release a resource that is assigned to a specific task.

default. A value, attribute, or option that is assumed when no alternative is specified by the user.

destination node. The node that provides application services to an authorized external user.

device control unit. A hardware device that controls the reading, writing, or displaying of data at one or more input/output devices or terminals.

device number. The unique number assigned to an external device.

device type. The general name for a kind of device; for example, 3330.

DFSMS. Data Facility Storage Management Subsystem.

direct access storage device (DASD). A device in which the access time effectively independent of the location of the data.

directory. (1) A type of file containing the names and controlling information for other files or other directories. Directories can also contain subdirectories, which can contain subdirectories of their own. (2) A file that contains directory entries. No two directory entries in the same directory can have the same name. (POSIX.1). (3) A file that points to files and to other directories. (4) An index used by a control program to locate blocks of data that are stored in separate areas of a data set in direct access storage.

display console. In MVS, an MCS console whose input/output function you can control.

DLL filter. A filter that provides one or more of these functions in a dynamic load library - `init()`, `prolog()`, `process()`, `epilog()`, and `term()`. See `cfilter.h` and `cfilter.c` in the `/usr/lpp/Printsrv/samples/` directory for more information. See also `filter`. Contrast with `DLL filter`.

DOM. An MVS macro that removes outstanding WTORs or action messages that have been queued to a console end-of-tape-marker. A marker on a

magnetic tape used to indicate the end of the permissible recording area, for example, a photo-reflective strip a transparent section of tape, or a particular bit pattern.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

double-byte character set (DBCS). A set of characters in which each character is represented by a two-bytes code. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires two bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with single-byte character set.

drain. Allowing a printer to complete its current work before stopping the device.

E

entry area. In MVS, the part of a console screen where operators can enter commands or command responses.

EMIF. ESCON Multiple Image Facility.

Enterprise Systems Connection (ESCON). A set of products and services that provides a dynamically connected environment using optical cables as a transmission medium.

EPDM. IBM SystemView Enterprise Performance Data Manager/MVS.

ESCD. ESCON Director.

ESCM. ESCON Manager. The licensed program System Automation for OS/390 includes all of the function previously provided by ESCM.

ESCON. Enterprise Systems Connection.

ETR. External Time Reference. See also Sysplex Timer.

extended MCS console. In MVS, a console other than an MCS console from which operators or programs can issue MVS commands and receive messages. An extended MCS console is defined through an OPERPARM segment.

F

FMID. Function modification identifier. The IBM release-specific product identifier such as HJE6610 for OS/390 Release 1 JES2.

FOR. File-owning region.

frame. For a System/390 microprocessor cluster, a frame contains one or two central processor complexes (CPCs), support elements, and AC power distribution.

FSS. functional subsystem. An address space uniquely identified as performing a specific function related to the JES. An example of an FSS is the program Print Services Facility that operates the 3800 Model 3 and 38xx printers.

functional subsystem (FSS). An address space uniquely identified as performing a specific function related to the JES.

functional subsystem application (FSA). The functional application program managed by the functional subsystem.

functional subsystem interface (FSI). The interface through which JES2 and JES3 communicate with the functional subsystem.

G

gateway node. A node that is an interface between networks.

generalized trace facility (GTF). Like system trace, gathers information used to determine and diagnose problems that occur during system operation. Unlike system trace, however, GTF can be tailored to record very specific system and user program events.

global access checking. The ability to allow an installation to establish an in-storage table of default values for authorization levels for selected resources.

global resource serialization. A function that provides an MVS serialization mechanism for resources (typically data sets) across multiple MVS images.

global resource serialization complex. One or more MVS systems that use global resource serialization to serialize access to shared resources (such as data sets on shared DASD volumes).

group. A collection of RACF users who can share access authorities for protected resources.

GTF. Generalized trace facility.

hardcopy log

H

hardcopy log. In systems with multiple console support or a graphic console, a permanent record of system activity.

hardware. Physical equipment, as opposed to the computer program or method of use; for example, mechanical, magnetic, electrical, or electronic devices. Contrast with software.

hardware configuration dialog. In MVS, a panel program that is part of the hardware configuration definition. The program allows an installation to define devices for MVS system configurations.

Hardware Management Console. A console used to monitor and control hardware such as the System/390 microprocessors.

HCD. Hardware Configuration Definition.

highly parallel. Refers to multiple systems operating in parallel, each of which can have multiple processors. See also n-way.

I

ICMF. Integrated Coupling Migration Facility.

IMS. Information Management System.

IMS DB. Information Management System Database Manager.

IMS DB data sharing group. A collection of one or more concurrent IMS DB subsystems that directly access and change the same data while maintaining data integrity.

IMS TM. Information Management System Transaction Manager.

initial program load (IPL). The initialization procedure that causes an operating system to begin operation.

instruction line. In MVS, the part of the console screen that contains messages about console control and input errors.

internal reader. A facility that transfers jobs to the job entry subsystem (JES2 or JES3).

IOCDs. Input/output configuration data set.

IOCP. Input/output configuration program.

IODF. Input/output definition file.

IPL. Initial program load.

IRLM. Internal resource lock manager.

ISPF. Interactive System Productivity Facility.

J

JES common coupling services. A set of macro-driven services that provide the communication interface between JES members of a sysplex. Synonymous with JES XCF.

JESXCF. JES cross-system coupling services. The MVS component, common to both JES2 and JES3, that provides the cross-system coupling services to either JES2 multi-access spool members or JES3 complex members, respectively.

JES2. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

JES2 multi-access spool configuration. A multiple MVS system environment that consists of two or more JES2 processors sharing the same job queue and spool

JES3. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely-coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job queue.

JES3 complex. A multiple MVS system environment that allows JES3 subsystem consoles and MCS consoles with a logical association to JES3 to receive messages and send commands across systems.

job entry subsystem (JES). A system facility for spooling, job queuing, and managing the scheduler work area.

job separator page data area (JSPA). A data area that contains job-level information for a data set. This information is used to generate job header, job trailer or data set header pages. The JSPA can be used by an installation-defined JES2 exit routine to duplicate the information currently in the JES2 separator page exit routine.

job separator pages. Those pages of printed output that delimit jobs.

K

keyword. A part of a command operand or SYS1.PARMLIB statement that consists of a specific character string (such as NAME= on the CONSOLE statement of CONSOLxx).

L

LIC. Licensed Internal Code.

list structure. A coupling facility structure that enables multisystem applications in a sysplex to share information organized as a set of lists or queues. A list structure consists of a set of lists and an optional lock table, which can be used for serializing resources in the list structure. Each list consists of a queue of list entries.

lock structure. A coupling facility structure that enables applications in a sysplex to implement customized locking protocols for serialization of application-defined resources. The lock structure supports shared, exclusive, and application-defined lock states, as well as generalized contention management and recovery protocols.

logical partition (LP). A subset of the processor hardware that is defined to support an operating system. See also logically partitioned (LPAR) mode.

logically partitioned (LPAR) mode. A central processor complex (CPC) power-on reset mode that enables use of the PR/SM feature and allows an operator to allocate CPC hardware resources (including central processors, central storage, expanded storage, and channel paths) among logical partitions. Contrast with basic mode.

logical unit (LU). In SNA, a port through which an end user accesses the SNA network in order to communicate with another end user and through which the end user accesses the functions provided by system services control points (SSCPs).

logical unit type 6.2. The SNA logical unit type that supports general communication between programs in a cooperative processing environment.

loosely coupled. A multisystem structure that requires a low degree of interaction and cooperation between multiple MVS images to process a workload. See also tightly coupled.

LP. Logical partition.

LPAR. Logically partitioned (mode).

M

MAS. Multi-access spool.

master console. In an MVS system or sysplex, the main console used for communication between the operator and the system from which all MVS commands can be entered. The first active console with AUTH(MASTER) defined becomes the master console in a system or sysplex.

master console authority. In a system or sysplex, a console defined with AUTH(MASTER) other than the master console from which all MVS commands can be entered.

master trace. A centralized data tracing facility of the master scheduler, used in servicing the message processing portions of MVS.

MCS. Multiple console support.

MCS console. A non-SNA device defined to MVS that is locally attached to an MVS system and is used to enter commands and receive messages.

member. A specific function (one or more modules/routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

message processing facility (MPF). A facility used to control message retention, suppression, and presentation.

message queue. A queue of messages that are waiting to be processed or waiting to be sent to a terminal.

message text. The part of a message consisting of the actual information that is routed to a user at a terminal or to a program.

microprocessor. A processor implemented on one or a small number of chips.

mixed complex. A global resource serialization complex in which one or more of the systems in the global resource serialization complex are not part of a multisystem sysplex.

MP. Multiprocessor.

MPF. Message processing facility.

MPFLSTxx. The SYS1.PARMLIB member that controls the message processing facility for the system.

MRO. Multiregion operation.

multiple console support (MCS)

multiple console support (MCS). The operator interface in an MVS system.

multi-access spool (MAS). A complex of multiple processors running MVS/JES2 that share a common JES2 spool and JES2 checkpoint data set.

multiprocessing. The simultaneous execution of two or more computer programs or sequences of instructions. See also parallel processing.

multiprocessor (MP). A CPC that can be physically partitioned to form two operating processor complexes.

multisystem application. An application program that has various functions distributed across MVS images in a multisystem environment.

multisystem console support. Multiple console support for more than one system in a sysplex. Multisystem console support allows consoles on different systems in the sysplex to communicate with each other (send messages and receive commands)

multisystem environment. An environment in which two or more MVS images reside in one or more processors, and programs on one image can communicate with programs on the other images.

multisystem sysplex. A sysplex in which two or more MVS images are allowed to be initialized as part of the sysplex.

MVS image. A single occurrence of the MVS/ESA operating system that has the ability to process work.

MVS router. The MVS router is a system service that provides an installation with centralized control over system security processing.

MVS system. An MVS image together with its associated hardware, which collectively are often referred to simply as a system, or MVS system.

MVS/ESA. Multiple Virtual Storage/ESA.

MVSCP. MVS configuration program.

N

n-way. The number (n) of CPs in a CPC. For example, a 6-way CPC contains six CPs.

NIP. Nucleus initialization program.

NJE. Network job entry.

no-consoles condition. A condition in which the system is unable to access any full-capability console device.

nonstandard labels. Labels that do not conform to American National Standard or IBM System/370 standard label conventions.

nucleus initialization program (NIP). The stage of MVS that initializes the control program; it allows the operator to request last minute changes to certain options specified during initialization.

O

offline. Pertaining to equipment or devices not under control of the processor.

OLTP. Online transaction processing.

online. Pertaining to equipment or devices under control of the processor.

OPC/ESA. Operations Planning and Control.

operating system (OS). Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible.

operations log. In MVS, the operations log is a central record of communications and system problems for each system in a sysplex.

OPERLOG. The operations log.

OPERPARM. In MVS, a segment that contains information about console attributes for extended MCS consoles running on TSO/E.

OS/390. OS/390 is a network computing-ready, integrated operating system consisting of more than 50 base elements and integrated optional features delivered as a configured, tested system.

OS/390 Network File System. A base element of OS/390, that allows remote access to MVS host processor data from workstations, personal computers, or any other system on a TCP/IP network that is using client software for the Network File System protocol.

OS/390 UNIX System Services (OS/390 UNIX). The set of functions provided by the SHELL and UTILITIES, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the OS/390 operating system that allow users to write and run application programs that conform to UNIX standards.

P

parallel processing. The simultaneous processing of units of work by many servers. The units of work can be either transactions or subdivisions of large units of work (batch). See also highly parallel.

Parallel Sysplex. A sysplex that uses one or more coupling facilities.

partitionable CPC. A CPC that can be divided into 2 independent CPCs. See also physical partition, single-image mode, MP, side.

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

password. A unique string of characters known to a computer system and to a user, who must specify the character string to gain access to a system and to the information stored within it.

permanent data set. A user-named data set that is normally retained for longer than the duration of a job or interactive session. Contrast with temporary data set.

PFK. Program function key.

PFK capability. On a display console, indicates that program function keys are supported and were specified at system generation.

PFKTABxx. The SYS1.PARMLIB member that controls the PFK table settings for MCS consoles in a system.

physical partition. Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

physically partitioned (PP) configuration. A system configuration that allows the processor controller to use both central processor complex (CPC) sides as individual CPCs. The A-side of the processor controller controls side 0; the B-side of the processor controller controls side 1. Contrast with single-image (SI) configuration.

PR/SM. Processor Resource/Systems Manager.

Print Services Facility (PSF). The access method that supports the 3800 Printing Subsystem Models 3 and 8. PSF can interface either directly to a user's

application program or indirectly through the Job Entry Subsystem (JES) of MVS.

printer. (1) A device that writes output data from a system on paper or other media.

processor controller. Hardware that provides support and diagnostic functions for the central processors.

Processor Resource/Systems Manager (PR/SM). The feature that allows the processor to use several MVS images simultaneously and provides logical partitioning capability. See also LPAR.

profile. Data that describes the significant characteristics of a user, a group of users, or one or more computer resources.

program function key (PFK). A key on the keyboard of a display device that passes a signal to a program to call for a particular program operation.

program status word (PSW). A doubleword in main storage used to control the order in which instructions are executed, and to hold and indicate the status of the computing system in relation to a particular program.

pseudo-master console. A subsystem-allocatable console that has system command authority like that of an MCS master console.

PSW. Program status word.

R

RACF. See Resource Access Control Facility.

RAID. See redundant array of independent disk.

RAMAC Virtual Array (RVA) system. An online, random access disk array storage system composed of disk storage and control unit combined into a single frame.

read access. Permission to read information.

recording format. For a tape volume, the format of the data on the tape, for example, 18, 36, 128, or 256 tracks.

recovery. The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a log.

redundant array of independent disk (RAID). A disk subsystem architecture that combines two or more physical disk storage devices into a single logical device to achieve data redundancy.

remote operations. Operation of remote sites from a host system.

Resource Access Control Facility (RACF)

Resource Access Control Facility (RACF). An IBM-licensed program or a base element of OS/390, that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system and logging the detected accesses to protected resources.

restructured extended executor (REXX). A general-purpose, procedural language for end-user personal programming, designed for ease by both casual general users and computer professionals. It is also useful for application macros. REXX includes the capability of issuing commands to the underlying operating system from these macros and procedures. Features include powerful character-string manipulation, automatic data typing, manipulation of objects familiar to people, such as words, numbers, and names, and built-in interactive debugging.

REXX. See restructured extended executor.

RMF. Resource Measurement Facility.

rmode. Residency mode. A program attribute that can be specified (or defaulted) for each CSECT, load module, and load module alias. RMODE states the virtual storage location (either above 16 megabytes or anywhere in virtual storage) where the program should reside.

roll mode. The MCS console display mode that allows messages to roll off the screen when a specified time interval elapses.

roll-deletable mode. The console display mode that allows messages to roll off the screen when a specified time interval elapses. Action messages remain at the top of the screen where operators can delete them.

routing. The assignment of the communications path by which a message will reach its destination.

routing code. A code assigned to an operator message and used to route the message to the proper console.

RVA. See RAMAC Virtual Array system.

S

SCDS. Source control data set.

SDSF. System Display and Search Facility.

shared DASD option. An option that enables independently operating computing systems to jointly use common data residing on shared direct access storage devices.

side. A part of a partitionable CPC that can run as a physical partition and is typically referred to as the A-side or the B-side.

single point of control. The characteristic a sysplex displays when you can accomplish a given set of tasks from a single workstation, even if you need multiple IBM and vendor products to accomplish that particular set of tasks.

single system image. The characteristic a product displays when multiple images of the product can be viewed and managed as one image.

single-image (SI) mode. A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with physically partitioned (PP) configuration.

single-system sysplex. A sysplex in which only one MVS system is allowed to be initialized as part of the sysplex. In a single-system sysplex, XCF provides XCF services on the system but does not provide signalling services between MVS systems. See also multisystem sysplex, XCF-local mode.

SLR. Service Level Reporter.

small computer system interface (SCSI). A standard hardware interface that enables a variety of peripheral devices to communicate with one another.

SMF. System management facilities.

SMP/E. System Modification Program Extended.

SMS. Storage Management Subsystem.

SMS communication data set. The primary means of communication among systems governed by a single SMS configuration. The SMS communication data set (COMMDS) is a VSAM linear data set that contains the current utilization statistics for each system-managed volume, which SMS uses to help balance space usage among systems.

SMS configuration. The SMS definitions and routines that the Storage Management Subsystem uses to manage storage.

SMS system group. All systems in a sysplex that share the same SMS configuration and communications data sets, minus any systems in the sysplex that are defined individually in the SMS configuration.

software. (1) All or part of the programs, procedures, rules, and associated documentation of a data processing system. (2) Contrast with hardware. A set of programs, procedures, and, possibly, associated documentation concerned with the operation of a data processing system. For example, compilers, library

routines, manuals, circuit diagrams. Contrast with hardware.

spanned record. A logical record contained in more than one block.

status-display console. An MCS console that can receive displays of system status but from which an operator cannot enter commands.

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volume or tape volumes, or a group of DASD, optical, or tape volumes treated as single object storage hierarchy. See tape storage group.

storage management. The activities of data set allocation, placement, monitoring, migration, backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automates these processes for you, while optimizing storage resources. See also Storage Management Subsystem.

Storage Management Subsystem (SMS). A DFSMS/MVS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

storage subsystem. A storage control and its attached storage devices. See also tape subsystem.

structure. A construct used by MVS to map and manage storage on a coupling facility. See cache structure, list structure, and lock structure.

subsystem-allocatable console. A console managed by a subsystem like JES3 or NetView used to communicate with an MVS system.

subsystem interface (SSI). An MVS component that provides communication between MVS and JES.

supervisor call instruction (SVC). An instruction that interrupts a program being executed and passes

control to the supervisor so that it can perform a specific service indicated by the instruction.

support element. A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

SVC routine. A control program routine that performs or begins a control program service specified by a supervisor call instruction.

symmetry. The characteristic of a sysplex where all systems, or certain subsets of the systems, have the same hardware and software configurations and share the same resources.

synchronous messages. WTO or WTOR messages issued by an MVS system during certain recovery situations.

SYSLOG. The system log data set.

sysplex. A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. See also MVS system, Parallel Sysplex.

sysplex couple data set. A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All MVS systems in a sysplex must have connectivity to the sysplex couple data set. See also couple data set.

Sysplex Timer. An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the MVS generic name for the IBM Sysplex Timer (9037).

system control element (SCE). Hardware that handles the transfer of data and control information associated with storage requests between the elements of the processor.

system console. In MVS, a console attached to the processor controller used to initialize an MVS system.

system log (SYSLOG). In MVS, the system log data set that includes all entries made by the WTL (write-to-log) macro as well as the hardcopy log. SYSLOG is maintained by JES in JES SPOOL space.

system management facilities (SMF). An optional control program feature of OS/390 and MVS that provides the means for gathering and recording information that can be used to evaluate system usage.

System Modification Program Extended (SMP/E). In addition to providing the services of SMP, SMP/E consolidates installation data, allows more flexibility in selecting changes to be installed, provides a dialog

Systems Network Architecture (SNA)

interface, and supports dynamic allocation of data sets.

Systems Network Architecture (SNA). A description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of networks.

system trace. A chronological record of specific operating system events. The record is usually produced for debugging purposes.

T

temporary data set. A data set that is created and deleted in the same job.

terminal. A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a link.

terminal user. In systems with time-sharing, anyone who is eligible to log on.

tightly coupled. Multiple CPs that share storage and are controlled by a single copy of MVS. See also loosely coupled, tightly coupled multiprocessor.

tightly coupled multiprocessor. Any CPU with multiple CPs.

Time Sharing Option (TSO). An option on the operating system; for OS/390 the option provides interactive time sharing from remote terminals.

TOR. Terminal-owning region.

transaction. In APPC/MVS, a unit of work performed by one or more transaction programs, involving a specific set of input data and initiating a specific process or job.

transaction program (TP). For APPC/MVS, any program on MVS that issues APPC/MVS or CPI Communication calls, or is scheduled by the APPC/MVS transaction scheduler.

U

undelivered message. An action message or WTOR that cannot be queued for delivery to the expected console. MVS delivers these messages to any console with the UD console attribute in a system or sysplex.

uniprocessor (UP). A CPC that contains one CP and is not partitionable.

UP. Uniprocessor.

V

VM. Virtual Machine.

virtual telecommunications access method (VTAM). A set of programs that maintain control of the communication between terminals and application programs running under DOS/VS, OS/VS1, and OS/VS2 operating systems.

volume. (1) That portion of a single unit of storage which is accessible to a single read/write mechanism, for example, a drum, a disk pack, or part of a disk storage module. (2) A recording medium that is mounted and demounted as a unit, for example, a reel of magnetic tape, a disk pack, a data cell.

volume serial number. A number in a volume label that is 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 data set on the volume.

VSAM. Virtual Storage Access Method.

VTAM. Virtual Telecommunications Access Method.

VTOC. Volume table of contents.

W

wait state. Synonymous with waiting time.

waiting time. (1) The condition of a task that depends on one or more events in order to enter the ready condition. (2) The condition of a processing unit when all operations are suspended.

WLM. MVS workload management.

wrap mode. The console display mode that allows a separator line between old and new messages to move down a full screen as new messages are added. When the screen is filled and a new message is added, the separator line overlays the oldest message and the newest message appears immediately before the line.

write-to-log (WTL) message. A message sent to SYSLOG or the hardcopy log.

write-to-operator (WTO) message. A message sent to an operator console informing the operator of errors and system conditions that may need correcting.

write-to-operator-with-reply (WTOR) message. A message sent to an operator console informing the operator of errors and system conditions that may need correcting. The operator must enter a response.

WTL message. Write-to-log message

WTO message. Write-to-operator message

WTOR message. Write-to-operator-with-reply message.

X

XCF. Cross-system coupling facility.

XCF PR/SM policy. In a multisystem sysplex on PR/SM, the actions that XCF takes when one MVS system in the sysplex fails. This policy provides high

availability for multisystem applications in the sysplex.

XCF-local mode. The state of a system in which XCF provides limited services on one system and does not provide signalling services between MVS systems. See also single-system sysplex.

XRF. Extended recovery facility.

IBM Redbooks evaluation

ABCs of OS/390 System Programming Volume 1
SG24-5597-00

Your feedback is very important to help us maintain the quality of IBM Redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.ibm.com/>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?

Customer **Business Partner** **Solution Developer** **IBM employee**
 None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

SG24-5597-00
Printed in the U.S.A.

ABCs of OS/390 System Programming Volume 1

SG24-5597-00

